THE ARGO PROJECT:
Machine vision based motion capture for tracking the trajectory of the pose of a mobile rigid body.

Geoffrey Allan Liggins

B. A. Sc. University of British Columbia, 1993

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

DEPARTMENT OF MECHANICAL ENGINEERING

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

April, 1998

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely availabe for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this for financial gain shall not be allowed without my written permission.

Department of Mechanical Engineering

The University of British Columbia

2324 Main Mall

Vancouver, B. C.

Canada

V6T 1Z4

Date:

April 23/98

Abstract

The objective of the Argo Project is to develop a tool that will track in real-time the motion of unconstrained, self-propelled, model ships in seakeeping tests done in towing tanks and manoeuvring basins. To meet the unconstrained requirement, the tracking system must be non- contact and can not interfere with the operation or motion of the model ship. An additional operating requirement is that the sensor must cover an area in excess of thirty square metres. An optical based sensor was selected as it satisfied these constraints.

Tracking the motion of the model ship is achieved with a predictive, extended Kalman filter (EKF), using feature point extraction from multiple synchronized images. The EKF is used because it can readily integrate and filter multiple noisy data sets. As well, it can generate an estimate of the pose, namely the position and orientation, of the model ship relative to the reference frame of the test tank. While this project is focused on ship tracking there are many other applications for a system of this kind.

The system under development makes use of the Qualisys™ camera and video processor hardware that extract image feature points and return them to a host computer. The incoming image feature points are then fed into tracking software developed in MATLAB™. The tracking software uses estimates of the image to do feature point correspondence and sorts the incoming data vector into the expected order. The sorted data vector is then used as the input vector for the EKF which computes the photogrammetric equations and computes the state vector for the pose of the mobile object being tracked.

This work is being undertaken at the University of British Columbia (UBC), Deparment of Mechanical Engineering, Maritime Engineering and Naval Architecture Research Laboratory. The organizaitons that assisted in this research effort are the Centre for Cold Ocean Resources Engineering, Intelligent Systems Group (C-CORE) and the National Research Council - Institute for Marine Dynamics (NRC-IMD).

Table of Contents

List of Tables

List of Figures

List of Abbreviations and Nomenclature

Supporting Organizations

UBC-ME	University of British Columbia, Department of Mechanical Engineering
C-CORE	Centre for Cold Ocean Resources Engineering
MUN	Memorial University of Newfoundland
MUN-OERC	Memorial University of Newfoundland, Ocean Engineering Research Centre
NRC	National Research Centre of Canada
NRC-IMD	National Research Centre of Canada, Institute for Marine Dynamics
NSERC	National Science and Engineering Research Council of Canada

General terms

CG	Centre of Gravity
DLT	Direct Linear Transformation
EKF	Extended Kalman Filter
IR	Infra Red
CCD	Charge Coupled Device
LED	Light Emitting Diodes
PSD	Position Sensitive Detector
PSP	Position Sensitive Photodiode

ARGO Project terms

MRB	Mobile Rigid Body
RB	Rigid Body
NDOF	number of degrees of freedom, default = 6
NORD	numerical order of kinematics model, default = 3
M	total number of cameras used in system

m	camera index number
N	total number of markers on target frame
n	marker index number
NM	number of observed markers in a image
nm	observed markers index number
u_n^m	horizontal image coordinate in camera m for marker n
v_n^m	vertical image coordinate in camera m for marker n
L_{1-11}^m	DLT parameters for camera m

Coordinate frame terms

Note: All reference frame coordinate systems refered to in this document follow the right handed convention unless other specifically noted.

Positions

(x_B^A, y_B^A, z_B^A)	position of reference frame B in the coordinate system of reference frame A
(x_n^W, y_n^W, z_n^W)	position of marker n in the *W*orld coordinate system
(x_C^W, y_C^W, z_C^W)	position of a *C*amera reference frame in the *W*orld coordinate system
(x_n^C, y_n^C, z_n^C)	position of marker n in a generic *C*amera coordiante system
(x_n^m, y_n^m, z_n^m)	position of marker n in the coordiante system for camera m
(x_n^T, y_n^T, z_n^T)	position of marker n in the *T*arget coordinate system
$(x_n^{MRB}, y_n^{MRB}, z_n^{MRB})$	position of marker n in the *M*obile *R*igid *B*ody coordinate system
$(x_{MRB}^W, y_{MRB}^W, z_{MRB}^W)$	position of the *M*obile *R*igid *B*ody reference frame in the *W*orld coordinate system
$(x_T^{MRB}, y_T^{MRB}, z_T^{MRB})$	position of *T*arget reference frame in the *M*obile *R*igid *B*ody coordinate system

Orientations

ψ_B^A	roll angle of reference frame *B* about the x axis of reference frame *A*
θ_B^A	pitch angle of reference frame *B* about the about y axis of the reference frame *A*
ϕ_B^A	yaw angle of reference frame *B* about z axis of the reference frame *A*
$\psi_C^W, \theta_C^W, \phi_C^W$	orientation of a *C*amera reference frame with respect to the *W*orld coordinate frame
$\psi_m^W, \theta_m^W, \phi_m^W$	orientation of camera reference frame *m* with respect to the *W*orld coordinate frame
$\psi_{MRB}^W, \theta_{MRB}^W, \phi_{MRB}^W$	orientation of the *M*obile *R*igid *B*ody reference frame with respect to the *W*orld coordinate frame

Coordinate system transformations

${}_B RPYT_A$	Roll-Pitch-Yaw-Translation homogeneous transformation from reference frame *A* to reference frame *B*
${}_B YPRT_A$	Yaw-Pitch-Roll-Translation homogeneous transformation from reference frame *A* to reference frame *B*
${}_B H_A$	homogenous transformation from reference frame *A* to reference frame *B*

Vectors

$\bar{V}_B^A$	vector describing pose of reference frame *B* with respect to reference frame *A*
$\bar{V}_{B\|C}^A$	vector from reference frame or point *B* to reference frame or point *C* with respect to reference frame *A*

Kalman Filter Terms

q	number of state variables = $NORD \cdot NDOF = 18$
r	number of estimated observations = $2 \cdot M \cdot N$
k	discrete time index ($k \in$ integer set)
I	Identity matrix
α	linear system dynamics model for single DOF [*NORD* x *NORD*]
A_k	linear system dynamics model [q x q]
β	variance matrix of system noise for single DOF [*NORD* x *NORD*]
C_k	linearized observation model [r x q]
$c_k(s_k, k)$	nonlinear observation function [r x 1]
e_k	observation-estimatation error [r x 1]
$\hat{e}_k$	estimated observation-estimatation error [r x 1]
e_k^*	state estimation error [q x 1]
g_k	observation vector based on actual state [r x 1]
$\hat{g}_k$	estimated observation vector based on state estimate [r x 1]
$\hat{g}_k^\dagger$	estimated observation vector based on preliminary state estimate [r x 1]
$\tilde{g}_k$	observed data vector (incoming image coordinates vector) [NM x 1]
K_k	Kalman gain matrix [q x r]
Γ	error covariance matrix for a single DOF [*NORD* x *NORD*]
P_k	estimate (error) covariance matrix [q x q]
$P_k^\dagger$	intermediate estimation filter covariance [q x q]
Q_k	variance matrix of system noise vector [q x q]
R_k	variance matrix of observation noise vector [r x r]
s_k	actual state vector [q x 1]

$\hat{s}_k$	estimated state vector [q x 1]
$\hat{s}_k^{\dagger}$	preliminary state estimate [q x 1]
ξ_k	observation noise vector [r x 1]
$\xi_{n_H}^m$, $\xi_{n_V}^m$	horizontal and vertical observation noise for camera n marker m
ω_k	system noise vector [q x 1]
$\omega_{1..6}$	system noise for a given degree of freedom
τ	sample period
$E\langle\gamma\rangle$	expected value of γ

Acknowledgements

I thank Dr. Sander Calisal, my supervisor, in the Department of Mechanical Engineering, University of British Columbia (UBC-ME), for his support during the course of this research.

I thank Dr. Ray Gosine of the Centre for Cold Ocean Resources Engineering (C-CORE) and Memorial University of Newfoundland (MUN) Faculty of Engineering and Applied Scence, for supervision and sponsoring my work. I greatly appreciate the opportunity to work with a world class organization such as C-CORE.

I thank Dr. Elizabeth Croft of the Department of Mechanical Engineering, University of British Columbia (UBC-ME) for her patience, her valuable advice regarding the Kalman filter, and her editorial input.

I thank Mr. Mike Sullivan of the National Research Council of Canada Institute for Marine Dynamics (NRC-IMD) for his assistance with the experimental phase of the project and his willingness to discuss the problems associated with motion tracking of floating objects.

I thank the Ocean Engineering Research Centre (OERC) at Memorial University of Newfoundland (MUN-OERC) for use of their target tracking video system hardware for the trials at the NRC-IMD.

I thank Mr. Ronan Oger for his support and the many discussions we had. I would also like to include my family and friends for their encouragement that made this possible.

I thank Mr. Luca Filipozzi, Mr. Tim Lam, and Mr. Allan Liggins, my father, for thier support and editorial input.

This work was done in cooperation with the Centre for Cold Ocean Resources Engineering, Intelligent Systems Group (C-CORE) and with the National Research Council of Canada, Institute for Marine Dynamics (NRC-IMD). Partial financial support for this work was provided through a of NSERC research grant held by Dr. Gosine. The equipment used in the experimental work belongs to NRC-IMD, C-CORE, MUN-OERC and UBC-ME.

Preface

The Argo Project is named after the constellation Argo visible in the southern hemisphere. This collection of points of light represent Jason's ship Argo, from Greek mythology. The Argo Project tracks the rigid body motion of a ship model in manoeuvring tests by observing reflective markers, namely points of light, strategically placed on the model.

In memory of Gordon Liggins.

Chapter 1: Introduction

The Argo Project explores the problem of quantitatively assessing the low frequency, large amplitude rigid body motions of an untethered model ship in manoeuvring and seakeeping tests. The Argo Project makes use of video based motion capture technology and the extended Kalman filter (EKF) to track the trajectory of the pose, namely the position and orientation, of a model ship in the reference frame of a test tank.

1.1 Motivation

Life aboard small commercial and passenger vessels in heavy seas is uncomfortable and dangerous to passengers, crew and vessel alike. Passenger vessels, ferries, pleasure craft, and fishing vessels must have good ride characteristics (normally referred to as seakeeping characteristics) in order to adequately provide comfort for passengers and safe working platforms for crew. Even more important than seakeeping properties is the safety of vessels in rough seas. The September 28th, 1994 [BNS 94] sinking of the passenger ferry Estonia off the Swedish coast illustrates the potential dangers of sea transportation. Every year in British Columbia, Canada, several vessels are lost due to rough sea conditions. Despite these hazards, the dynamics of vessels in rough seas are not fully understood. Proper assessment of the seakeeping properties through model tests allows designers to evaluate potential hullforms and if necessary make any changes to the vessel prior to construction.

Traditional seakeeping test protocols have the model ship pushed forward along the length of a towing tank in a head sea. During these tests the model ships are instrumented to record heave, pitch, drag force and vertical accelerations at the bow and centre of gravity (CG). Figure 1.1 illustrates the definitions of ship motion terms. To allow for instrumentation of the model ship, its motions are typically constrained to move only in heave, pitch and occasionally surge. The

assumption is made that roll, yaw and sway are minor motions and are held fixed. For quartering, beam and following seas this assumption fails.

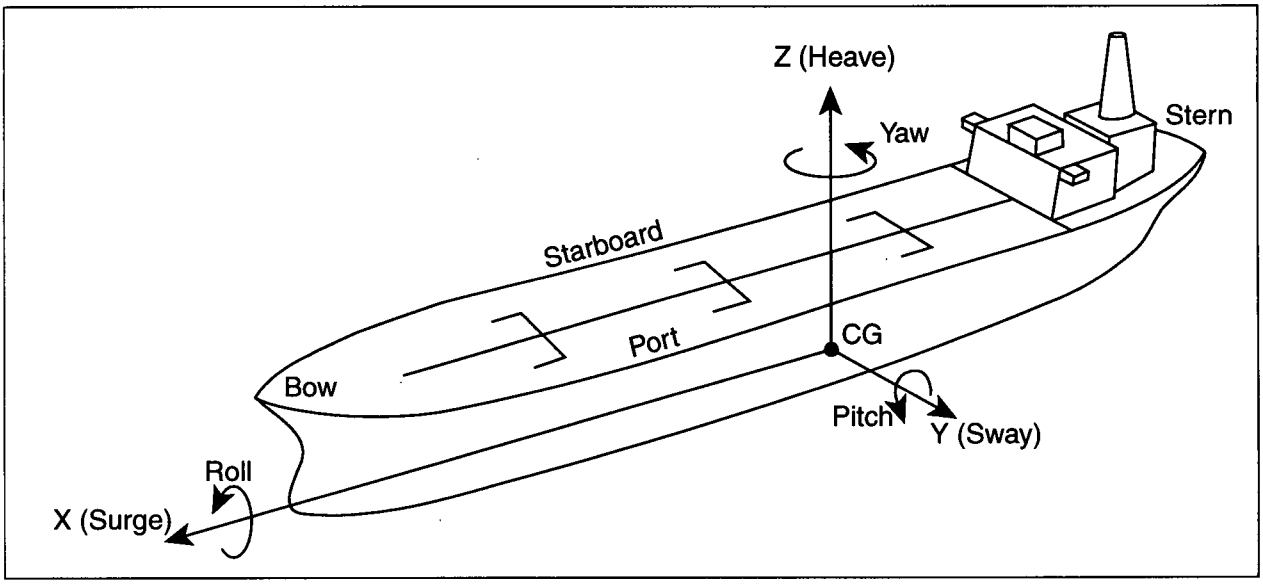


Figure 1.1: Ship's coordinate frame and terminology

Using free running, unconstrained model ships and floating bodies, investigators can achieve a closer representation of a ship's response to wind, wave and current excitation in model scale. The requirements of the system to measure the motion of an unconstrained floating body in a manoeuvring tank are:

- must be non-contact so as not to constrain the floating body.
- must be capable of being used near water.
- must cover a large test volume (30 m X 30 m X 2 m).
- must not cause any electromagnetic interference with other sensors or data acquisition systems being used.
- must measure large amplitude, low frequency (< 10 Hz) motions.
- must have sufficient resolution and accuracy to yield useful data.
- must capture the six degree of freedom (6 DOF) motions of the floating body.
- must have a high signal to noise ratio.
- must have a resolution of 1:5000 or better.

The intended use of the data being collected is to gain insight into the motions crew members may experience on a full scale ship. The data may be used to evaluate different designs in comparative studies. The data could also be used in a feed back loop if the investigators were testing heading control in different sea states.

Of the many non-contact motion measurement systems currently available, the video/cine photogrammetry method is one of the very few to meet the above requirements. Video/cine photogrammetry, an established technique in fields such as biomechanics, is a method for extracting motion from a sequence of images by extracting dimensional information from individual images from single and multiple points of view.

1.2 Motion Tracking

The problem of using a sequence of images, either film or digital, to analyse motion is a well studied problem that has been approached using a variety of methods. The problem is divided into two separate areas: (*i*) photogrammetry, the science of extracting dimensional information from the images, and (*ii*) tracking, i.e, quantifying the motion. The system presented in this thesis tracks the motion of a target composed of a set of markers that is attached to a model ship. In generic terms this is a mobile rigid body (MRB). The pose (position and orientation) of the MRB can be monitored because pose of the target is known with respect to a coordinate frame of the MRB, which, in the case of a model ship, has an origin at the centre of gravity (CG).

Systems similar in function to the one being presented are already being used in ocean engineering test facilities worldwide to study the motion of ship models and models of other floating structures [MARINTEK 97, IMD 5 & 6, & Sirehna 97]. At least two such systems are in use at NRC-IMD, where the most current system has been developed in house [Sullivan 93 & 97]. This system has been used to monitor wave excited motions for a variety of platforms, including, but not limited to, life rafts, aqua-culture pens, a tension leg platform and ship models. The work presented in this thesis is not intended to duplicate the system designed at NRC-IMD, but rather to explore and evaluate an alternate method that can be used in conjunction with it.

The proposed system differs from the existing NRC-IMD system by being able to track an MRB with a minimum of one video camera to a maximum limited only by the hardware required to interface the cameras to the host computer. Also, the MRB does not have to remain in the field

of view of all cameras in the system, as long as it remains in the view of at least one. In effect the proposed system requries a minimum amount of data, but can make use of all the data made available to it. On the other hand, the NRC-IMD system is a typical two camera stereo system where the MRB must remain in the field of view of both cameras for it to maintain tracking the target object. The advantages of going to a redundant multi-video-camera system is that it increases the test volume while maintaining resolution and can increase the accuracy and reliablilty of the measurements made during the test.

1.3 Thesis Structure

This thesis is broken down into 7 chapters, including this introduction, and a set of appendices that contain additional details of the algorithms developed and used for this project in both simulation tests and the analysis of experimental data collected in a controlled motion study.

Chapter 2 discusses the selection of a machine vision approach to tracking the motions of a model ship, as well as existing vision based tracking systems that are either under development in research facilities or are commercially available. Special attention will be given to those systems that are relevant to this project.

Chapter 3 explains photogrammetry, the method of obtaining spatial information from images. The colinear equations, the fundamental set of equations that describe a pin hole camera positioned in space, along with the Direct Linear Transformation (DLT), a method of solving the colinear equations for use with non-metric, or uncalibrated off the shelf, cameras, are presented.

Chapter 4 presents the Extended Kalman Filter (EKF) and how it is implemented to with the photogrammetry equations to generate the state vector that describes the six degrees of freedom of the mobile object being tracked.

Chapter 5, discusses both the hardware and the software components of the implementation of the Argo system.

Chapter 6, presents the experimental procedure used to test the Argo system and the results

from those tests. Experiments include both computer simulation and physical testing.

Chapter 7, concludes the body of this thesis and discusses potential continuations of this work. This is followed by six appendices of supporting material.

Chapter 2: Current Motion Capture and Analysis Technology

2.1 Introduction

The noninvasive nature of photography and machine-vision sees it regularly employed to qualitatively and quantitatively evaluate the motion of a variety of systems. Motion capture is the collection of quantitative data related to the motion of an object or point of interest with respect to a particular frame of reference. Motion analysis involves four components: (*i*) the processing of motion capture data for tracking trajectories of points of interest; (*ii*) the assimilation of motion data with additional data from other sensors that monitor other states of the system; (*iii*) the comparison of data with previously acquired data for either repeatability or change; and, (*iv*) the determination of changes in position, orientation and the associated rates (velocity, acceleration, and jerk). The system to track the motion of a moving target is the combination of both motion capture and motion analysis. The motion capture-analysis system must be tailored to the specific motion being observed and the planned uses of the collected data. One of the goals of this research is devise a system that can track the motion of a model ship in a simulated seaway. The selection of an appropriate motion capture system is important to the success of the project. However, the main thrust of this project is to track and analyse the motion.

2.2 Selection of a Motion Capture System

Motion capture can be achieved with a variety of sensor systems that may be used separately or in concert, each with strengths and weaknesses. The sensor systems available for motion capture cover a wide range of working volumes from the very small to the very large, they also have a range of resolutions, accuracies, repeatabilities, response times, sample rates and unique environmental constraints and accommodations. The applicable sensor systems include: electro-mechanical sensors, inertial sensors and inclinometers, sonar, radar, electromagnetic, laser range scanning, differential global positioning system (DGPS), and optical imaging devices (machine-vision).

Selection of the proper sensor system must be dictated by the system under observation and the type of observations to be made.

2.2.1 Requirements for monitoring model ship motions

For this project the system being observed is a model ship experiencing pseudo-random excitation from waves in a simulated seaway. The ship is modelled as a rigid body; therefore, the system will be observing rigid body motion. The 6 degrees of freedom (DOF) of a model ship as illustrated in Figure 1.1 are: roll, pitch, yaw, surge, sway, and heave. The model should be free to move and be unconstrained by the data collection system. All investigators would like to have perfect accuracy[1], infinite resolution[2], and perfect repeatability[3] but, due to sensor limitations, this is not possible. For observing a model ship in a simulated seaway, a resolution of 2 mm over a range of 10 m and 0.1 degrees over a range of 30 degrees would be desirable with as much accuracy and repeatability as is possible.

The motions of a model ship are expected to have an amplitude and frequency corresponding to the exciting wave. For a one metre wave height the heave would be expected to be about one metre. Angular motions are expected to have an amplitude of 0-30 degrees. Typically, the response frequency of a model ship is on the order of 1 to 3 Hz. Under power, the frequency may increase if the rudder is being oscillated. The motion of the model ship should be sufficiently over sampled so that its motion and vibrations can be observed. For example ten times the expected frequency of 3 Hz would be 30 Hz. This far exceeds the Nyquist frequency constraint for observing dynamic systems and provide sufficient data to make the tracking algorithm converge and hold the track.

The model ship will be observed in either a towing tank or a manoeuvring basin. These facilities tend to be large and range in size from 50-200 m in length and 3-100 m in width. The test

1. The accuracy of a sensor is indicates the closeness of a measured value to the true value.
2. The resolution of a sensor indicates the smallest measurable incremental value.
3. Repeatability is the ability to reproduce the same measured value when duplicating a measurement.

volume should be considered to be in this range even if certain tests observe nearly stationary model ships, as in station keeping tests.

Towing tanks and manoeuvring basins are typically indoors and have a room temperature of 15°C, except for ice tanks which can be as low as -20°C. The tanks are water filled with wave makers located along the walls of manoeuvring basins and at the end of towing tanks. As a result of wave generation it should be expected that the areas in the vicinity of the tests will be splashed or wet. Every effort should be made to ensure that special data acquisition equipment is not splashed as water drops can interfere with the data collection process, especially for optical based systems. Also, computers and other electrical equipment could be permanently damaged if flooded. Testing that requires specific lighting conditions may be difficult to do because the facilites that house test tanks, typically have overhead mercury vapour or fluorescent lighting, that would normally remain on during testing for safety and observation reasons.

Safety considerations for the model are few except that the investigators would like to minimize damage to the model so that it can be used in future tests. The instrumentation and data acquisition equipment should be well protected because shock can effect calibration, accuracy, resolution, and may necessitate expensive repair or replacement. Safety of the investigators is paramount in any test procedure and therefore electrical connections should be made safe for a wet environment and all equipment should be securely fastened so that it can not fall into the water or injure anybody in the test area. Also, appropriate safety equipment should be made available during the test for a wet and possibly cold environment.

Creating a budget in a testing environment is always a difficult problem. A system such as the one being proposed would not be acquired for a single project; instead it would be amortized over several tests to make it economically viable. Any motion capture system capable of observing a model ship will have a large capital cost ranging from $20K to $250K depending on the facility and the type of tests currently under way and those planned for the future.

2.2.2 Evaluation of sensor systems

Sensor system selection must be dictated by the system under observation, what is the desired data, and in what form should the data be presented. To illustrate this point, in the case of monitoring the motions of a full scale ship a combination of DGPS, inertial systems and possibly radar would typically be used. However, these same sensor systems are inappropriate for monitoring the motions of a model ship. These two problems, although similar, are very different because they take place in different environments and have different scales.

A complete description of the motion capture problem is required before selection of the appropriate sensor system. Using a two step elimination process, all but the valid sensor systems are rejected. First, obviously incompatible sensor systems are rejected. Second, the published performance specifications of the remaining sensors are compared with the required specifications of the motion capture system as shown in Table 2.1.

Performance specifications and general descriptions of the functionality of different sensor systems are published in reference texts [Borenstien 96, deSilva 89 & Karara 89], papers [Mackay 96 & Mulder 94], trade journals such as *Vision Systems Design* by Pennwell Publication and *Photonics* by Laurin Publication, and product literature available from any manufacturer or distributor [CIDTEC 97, Dalsa 96, Qualisys 96 & Trimble 97]. As well there is vast amount of information available on the internet [Ariel 97 & Trimble 97].

Comparing the model scale ship testing requirements, summarized in the second column of Table 2.1, with the available sensor systems. The electro-mechanical sensors and the elector-magnetic sensors can be rejected because they physically constrain the model ship's motion. The DGPS system is rejected because it can not be used in an indoor environment. The time of flight systems: sonar and radar are rejected because the number of degrees of freedom of the sensor is too low and the signal to noise ratio is also low. The scanning laser range finder is rejected because

Table 2.1: Sensor System Comparison

Categories	Model Scale Ship Testing Requirements	Full Scale Ship Testing Requirements	Electro-Mechanical Sensors	Electro-Magnetic Sensors	TOF Sonar Radar	Scanning Laser Range Finder	DGPS	Inertial Systems	Machine Vision
Suitable for indoor use	Yes	No	Yes	Yes but environment sensitive	Yes but environment sensitive	Yes	No	Yes	Yes
Workspace size	small to large	unlimited	small & medium	small to medium	medium very large	medium to very large	unlimited	unlimited	very small to large
Can be used near water	Yes	Yes	Yes with precautions	Yes, but can cause false signals	Yes, but can cause false signals	Yes	Yes	Yes	Yes
Safety Concerns	minimal	minimal	minimal	EMF levels	EMF levels	LASER radiation	minimal	none	none
# of DOF of sensor	6	6	1-6	3	2	3	3	1-6	6
test object unconstrained	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
accuracy	1:1000	1:20	1:2500	1:1000	1:20	1:2000	2 - 5 m	1:1000	1:5000
resolution	1:5000	1:100	infinite	1:1500	1:100	1:6000	10 cm	1:10000	1:30000
bandwidth	30 Hz	10 Hz	up to 10 kHz	100 Hz	30 - 100 Hz	10 - 40 Hz	2 Hz	10 - 1000 Hz	30 - 1000 Hz
Signal to noise level	high	high	high	medium	low	medium	medium to high	medium	medium to high
Other			requires heavy infra-structure	sensitive to metal in test volume	prone to false or ghost signals	significant image processing	3 units required for 6 DOF	prone to drift	sensitive to lighting

Workspace Size: very small < 10 cm; small 10 cm - 1 m; medium 1 m - 3 m; large 3 m - 50 m; very large 50 m - 1000 m; unlimited > 1000 m

the number of degrees of freedom of the sensor are too few. Additionally the laser-based system's requirement of significant image processing eliminated it from contention. A machine-vision based systems was selected because its characteristics most closely match the requirements for testing a model ship.

Having gone through this process, the most appropriate sensor system for observing the motion of a model ship in a simulated seaway is a video-based tracking system possibly in combination with inertial and inclinometer sensors on board the model. The main reasons for this are these systems are non-contact, can cover a large test volume, can be used indoors, are safe to use near water with proper precautions, and they have sufficient accuracy, resolution, response time, and sampling rates. The on board sensors would be coupled with a master data acquisition system via a spread spectrum radio link, which could be shared with a remote control system. Due to the complexity of the overall problem, the added cost to the project and the fact that at this stage the project is a proof of concept only; the video-based motion capture was pursued and the inertial sensor system was left for future work.

Typical video tracking systems capture a set of synchronized image sequences of a 3D dynamic scene, from multiple points of view. From analysing both stationary and moving feature points, also known as landmarks, tokens or object points, in the images the motion of the scene relative to some fixed reference frame can be measured. There are many commercial, video based, close-range photogrammetry, motion capture systems that have been developed and are currently available on the market. These commercial systems are mainly used for recording human motions for medical, animation and sports applications. Few of these systems are optimized or even capable of tracking rigid body motion. Video tracking is also being actively pursued by the research community for non-contact measurement in experimental testing [Johnson 90, Rediers & Wysner 83], to improve human motion analysis [Romilly 95 & Safee-Rad 87] and to aid in robot navigation, servoing and obstacle avoidance [Borenstien 96 & Wilson 93].

2.2.3 Imaging Technology

Early implementations of motion capture based on image sequences was achieved through either cinematography [Shapiro 77], sequenced still photography, as Muybridge did in his study of a galloping horse in 1877 [Cook 81], or multiple exposure strobe photography [Edgerton 87]. In all of these cases the images of the scene were recorded on film with a camera or group of cameras synchronized together and with the action in the scene with the appropriate lighting.

Imaging technology used in film and newer electronic cameras is a very large topic and covers a wide range of techniques and equipment depending on such things as lighting conditions, the desired frame rate, the desired image resolution, scene environment such as temperature and humidity. Film has the highest resolution images and the highest frame rate for medium to high resolution applications. However, for the majority of applications it has been replaced by electronic imaging because it is easier to use and does not incur film and processing costs. Film has also lost favour because of the difficulty in digitizing a film image for computer based image processing. However, film is still the cheapest and most effective method for high resolution and high frame rate applications because of high costs for electronic imaging technologies that can approach the same image quality and higher frame rates of film.

The most common electronic imaging technology is the charged coupled device (CCD). It is low cost, and can be easily interfaced to a computer through a dedicated "frame grabber" input/output (I/O) digital signal processor (DSP) board for image processing. The CCD array sensor is most sensitive to the red and near infrared (NIR) end of the light spectrum and is available over a broad range of resolutions from 64 X 64 to 5000 X 5000 [Dalsa 96].

Scientific imaging typically has a resolution of 512 X 512. However, much of the commercially available motion capture systems are based on consumer and industrial CCD cameras, the most commonly available, which are interlaced composite video encoded with either PAL format, 625 video lines at 25 frames per second (fps), for most of Europe, or NTSC format, 525 video lines at

30 fps, for North America and Japan. Both formats have an image aspect ratio of 4(H):3(V) [Stremler 82]. To allow for sync-signals and field and line retraces, the working image size for North American camera systems is usually 640 X 480.

The majority of motion capture applications in both commercial and research efforts, use these commonly available consumer and industrial CCD based video cameras. For those systems that use infra-red (IR) markers, both passive and active ones, the camera lenses are usually fitted with an IR band pass filter. For the reflector type markers the cameras will have an IR flash mounted around the lens which is triggered by the frame rate of the camera [Motion 96, Qualisys 96 & Vicon 96]. To meet the demands of capturing higher speeds, such as a baseball pitcher's arm, newer systems are now using higher performance imaging and computer components, resulting in frame rates up to 1000 fps [Qualisys 96].

Some other electronic imaging devices, used for special applications where a CCD camera may not function properly, are the intensified charge coupled device (ICCD) for low light applications, silicon intensifier technology (SIT) for low light underwater imaging because of its sensitivity to blue light [Mackay 96] and the charge injection device (CID) for ultraviolet (UV) imaging and high radiation environments [CIDTEC 97]. The imaging tube is an older analogue method of capturing an image that has been replaced by CCD arrays for the majority of uses. However, there still remains some special applications for this older technology.

Another photo-optical sensor that can image a marker or landmark in the scene is a position sensing photodiode. This device, unlike the CCD array, can only view one marker at a time. A position sensing photodiode (PSP), also referred to as a position sensitive detector (PSD), is an analogue device that returns the position of a bright spot projected onto the sensor. If more then one light source is incident to the sensor surface the sensor will return an average position of the multiple bright spots. The spectral sensitivity of a PSP makes it ideally suited for red and near infrared light (NIR) sources. This device has a response rate 10-15 kHz, depending on the specific man-

ufacturer [On-Trak 97]. If the lens in front of the PSP sensor is fitted with a band pass filter that is matched with the marker light source wave length then the majority of stray light should be eliminated, minimizing noise in the signal. With two or more synchronized PSPs the position of the point of light can be triangulated similar to the method used in a conventional camera. To accommodate multiple feature points the IR light emitting diode (LED) markers are strobed in sequence, with only one lit at any given time.

2.2.4 Markers

Feature points are usually represented by markers with high contrast to the test subject and the scene background [Johnson 90]. Markers are used to speed the feature point extraction process but they are not required as detailed image processing can be used to identify natural points or an operator can manually select a point in the image.

One common approach is to use passive retroreflective markers coupled with an IR strobe. Another approach uses active IR LED markers that, when viewed through a filter matched to the wavelength of the marker, produces a high contrast black and white image where the markers are easily identified. The passive markers are typically spheres, half spheres or circular dots coated in retroreflective paint that is highly reflective in the IR regime. Spherical markers are preferred because the apparent centroid of a sphere is independent of viewing angle. With other shapes the centroid can appear to shift with viewing angle. For example, the centroid of a hemisphere can appear to move by 42% of the radius of the marker when viewed at the side as compared with the view from normal to the hemisphere [Gieck 90]. A difficulty of spherical IR reflectors is that all markers appear the same in the image and cannot be uniquely identified from the raw images. Thus, the tracking software must identify them. Stray reflections from equipment or the surface of a liquid, sometimes known as ghost markers, can appear in the data set and may confuse the tracking software and requiring manual intervention to complete the tracking task.

Active markers such as small light bulbs and LEDs provide strong signals and typically have a

smaller size than the reflective passive markers. The active markers can be sensitive to observation angle, particularly LEDs, therefore; for monitoring small displacements they are acceptable, for large displacements a cluster of LEDs that can be viewed from all possible viewing angles is required.

As with passive spherical reflective markers, small lights all appear the same in the images and a means of identifying them is required so that their positions can be triangulated. One approach used to identify individual LEDs is to strobe them in a sequence having only one lit at any one time. The advantage of this approach is that it is relatively fast, and since only one marker is illuminated at any given time correspondence between images and the marker is not an issue. A disadvantage of this method is that if the target moves appreciably over the duration of one sample iteration and each marker is illuminated once, then some measurement error will be introduced because the image is not a "snap shot" of the marker locations at one point in time, but rather a rolling sample. This method has been implemented in commercially-available motion capture-analysis systems [Charnwood 97, NDI 97, Selspot 96 & Yaman 97].

An advantage of using IR light, either with passive or active markers, is that overhead lighting need not be turned off during testing, making it safer for researchers to conduct the test. However, strong ambient levels of IR light can cause noise or clutter in the image making feature point extraction less reliable.

Many systems make use of the visible light spectrum and use markers that contrast from the scene. One popular method is to use high contrast black and white patterns that are relatively easy to extract from an image. Some systems have employed coloured and/or differing shaped markers to uniquely identify individual markers. Other systems use no markers and rely on the operator to define a feature point from the scene in the image. With no markers available for tracking, this approach requires either the operator to manually track the point of interest frame by frame or complex image processing and tracking algorithms that can follow these points of interest. As the level

of complexity of the point of interest increases, the requirements for more complex imaging hardware and image processing for detecting colour detail and/or shape of the feature point in the scene also increases. This added complexity would likely be reflected in the cost of the image capture system. As well, such systems are more sensitive to ambient light that may change the apparent colour of a point of interest or cast a shadow that may change its shape.

Obtaining the image location and size of what is perceived to be a marker, namely, "feature extraction" is done through image processing. This topic is not discussed in this thesis as it is outside of the scope of the project. The image processing done in the collection experimental data for this project was accomplished in firmware built into the Qualisys video processor hardware.

2.2.5 Commercial motion capture systems

There are many machine-vision based motion capture systems commercially available today. The majority of these systems have been designed for human motion analysis for kinesiology and medical studies, with a few notable exceptions. Recently the systems originally designed for the sports [Blackburn 96] and medical communities [Harris 96] are being used by the entertainment industry for animation in movie productions and action video games [Delaney 97]. These systems track individual markers attached to the patient, client or actor at strategically located points such as the hip, knee, elbow and shoulder. From this the operator can infer body movements from the trajectories of the markers attached to the test subject, captured with synchronized video cameras while the test subject performs the target physical activities [Harris 96]. Often these systems do not operate in real-time and rely on post-processing of recorded data. Unfortunately, very few of these systems are designed specifically to follow rigid body motion. Rigid body motions have additional constraints that allow for speedier and more accurate tracking, with greater robustness and effective resolution. This is not to say that these systems could not be used for tracking rigid body motion but rather that they are less then optimal.

Systems designed for human motion analysis track the 3 DOF of many feature points. On the

other hand, a system designed to track rigid body motion is intended to track the position of the origin and the orientation of a coordinate frame attached to the test object. The techniques used in these two applications are not entirely separate but different enough that software developed for kinesiology work is inappropriate to study the rigid body motions of ship models. Much of the software supplied with these systems is setup for clinical trials of patients or atheletes, and the only output available is of time histories of the motions of individual markers attached to the test subject. As well due to the proprietary nature of this type of software, it was not possible to obtain the necessary building blocks to adapt the existing program to track large-amplitude, low-frequency rigid body motion. Another problem with the proprietary nature of the software is that the motion analysis techniques used are not openly published forcing a researcher wanting to expand the system beyond its original intended use to guess at what the company has done or to build a secondary data post processing routine. As a result, system development for this application had to start from the ground up. Fortunately, many of the mathematical building blocks for developing a rigid body tracking system are available in the public domain.

Many of the manufacturers of this equipment make use of proprietary video processors with built in firmware that can not be modified by the user. The tracking routines or portions of them are usually included in this hardware forcing developers to rely on partial execution of the tracking process. This is a benefit because it will save time, but it is also a detriment as it limits what can be done and how much access one has to the raw, unmanipulated data. Before using such a system the developer will require a detailed knowledge of how the data is manipulated and of the output format so that they can link it into their work.

Some commercial systems, although not optimized to track large amplitude, low frequency rigid body motion, do so by tracking the individual markers that make up the target and then reconcile their motion to that of the mobile rigid body (MRB). Other systems have been designed specifically for tracking rigid body motion. These systems are typically configured for industrial inspec-

tion tasks or for integration with robotics for calibration, evaluation, and control input.

There are a number of the commercial systems designed for human motion studies that have been adapted to the problem of observing the motion of model ships. The U. S. Army Corps of Engineers is using a Motion Analysis Expert Vision system at their Waterways Experiment Station in Vicksburg, Mississippi. They use this system to follow model ships in their mock-up of the lower Mississippi [Motion 97]. The Oxford Metrics Vicon 370 system is being used at the Shipping and Marine Department at Strathclyde University in the United Kingdom for capture roll, pitch and yaw of a model in their towing tank [Vicon 96]. Both Marintek of Norway and NRC-IMD have employed both the Selspot and OPTOPOS (Optical Positioning) systems to track the rigid body motion of model ships in manoeuvring tests [MARINTEK 97 & Selspot 96]. At NRC-IMD the OPTOPOS system, believed to have been made by Saab, is no longer operational. Presently NRC-IMD, MUN-OERC and NRC-Hydraulics of Ottawa, Ontario are using a system developed at NRC-IMD based on the Qualisys Mac Reflex™ system [Sullivan 93 & 97]. This system has been proven in comparative tests and is routinely used on commercial and internal research projects.

For further information on specific systems the reader is directed to tables A.2, A.3, and A.4 in Appendix A: Motion Capture Systems for a comparative review of some of the commercially-available machine vision based motion capture systems.

2.2.6 Selected motion capture system

The Qualisys MacReflex™ camera and video processor hardware with its built in firmware was selected for the Argo Project because it met the requirements for observing a model ship in a tank and was compatible with other on-going research projects at NRC-IMD. This system has a suitable sample rate, sufficient resolution and accuracy, does not constrain the model, can be used with overhead lighting, and has built in frame grabbers and image processing for extracting the image feature points.

Only the hardware component of the Qualisys MacReflex™ system is being used because the

software normally supplied with the system can not track the rigid body pose of one or more MRBs in real-time. The original software is designed for doing human motion analysis and due to its proprietary nature could not be modified to meet the needs of the project. NRC-IMD are using software developed in house to acquire data from the Qualisys hardware and track the pose of a model ship [Sullivan 93 & 97]. The Argo Project is currently concentrated on the tracking algorithm and how it can be improved; therefore, it relies on the NRC-IMD software to capture and record data gathered with the Qualisys hardware.

2.3 Motion Analysis Techniques

Motion analysis from a sequence of images is the extraction and interpretation of motion from those images. This data reduction process includes photogrammetry, the extraction of position data from an image, coupled with tracking and data fusion of image coordinates from multiple image sequences.

Early motion-capture-analysis projects that went beyond a purely qualitative assessment used metric cameras and an analogue electro-mechanical-optical device, known as a stereoplotter, to manually digitize film sequences frame by frame. The manually digitized image feature point centroids were then entered into a computer for triangulation of each point. From the triangulation results the pose of the object would be resolved using a least squares fit for each frame. With the improvement of imaging, electronics and computing ability the above described process has evolved to become more efficient, faster and less costly but essentially remains the same. The stereoplotter is replaced with a frame grabber and automated feature point location software. Entire sequences of images can now be processed without any manual intervention by the operator. A benefit to increasing the speed and the ability to automate the process allows for the real-time tracking of motion for control applications.

The majority of the information, published on the topic of motion analysis based on a sequence of images, comes from the reference texts and the university and government research community

as private companies tend to hold this information as a trade secrets to prevent their competition from profiting from their efforts. As a result little is openly published about the techniques and methods used in the commercial systems discussed in sections 2.2.5, 2.2.6.

2.3.1 Photogrammetry

Photogrammetry is a technique for extracting three dimensional (3D) information scene from two dimensional (2D) images of the scene. Photogrammetry is regularly used for mapping, surveying and measurement of static and dynamic systems. It is used when physical measurements can not be performed because they may damage or interfere with the system under observations, or when the range and scale of the scene make measurement impractical or impossible. Photogrammetry makes use of the geometric relationship between 3D space to the 2D representation in the image plane. This transformation is essentially a plane perspective projection with a few additional terms for correcting lens and image plane distortions. The methods and techniques of photogrammetry are well developed but some advances have been made coupled with advancing technology [Karara 89]. The concept of motion analysis from a sequence of images is an extension of traditional photogrammetry to include a time vector [Faugeras 93].

The geometric relationship between 3D space and the 2D image plane is typically referred to as the camera model. The simplest camera model, typically the basis for more detailed ones, is the pin hole camera model. This model is the plane perspective projection model where the feature point in the scene, the focal point and the image point all lie on the same vector in space. A parameterized version of this basic camera model is called the Direct Linear Transformation (DLT). This model was designed for non-metric commercial off the shelf (COTS) cameras for still photogrammetry [Aziz 74 & Marzan 76]. Triangulation of feature points in the scene, as well as calibration are simple, straight-forward procedures. Although the DLT was developed for still photogrammetry it has been successfully applied to motion analysis at both the research [Anglin 93, Miller 80, Safee-Rad 87 & Shapiro 77] and the commercial [Areil 97& Peak 96] level. The

DLT has been selected as the camera model for the Argo Project.

As a single 2D image of a 3D scene does not contain enough information to determine a 3D position of a point in the scene, additional information is required. Typically this extra information comes in the form of one or more additional images taken from different points of view, covering the viewing volume. This is the stereo vision approach. An added benefit is improved resolution of the observation system as the individual cameras that make up a multiple camera system may suffer from low resolution problems along the focal axis of the camera. Using the principles of triangulation and knowing the pose of the different cameras in the world coordinate frame, the 3D position of any point can be determined from images of the scene.

Systems that rely on triangulation have difficulties when a marker is occluded from view and is not visible in at least two fields of view. In this case, the position of that marker is not computed at all.

Once the 3D position of feature points fixed relative to the coordinate frame of the MRB has been determined, the pose of the MRB can be found. The pose of the MRB can be found by matching the known positions of the feature points in the coordinate frame of the MRB with the positions of those same feature points measured in the world coordinate frame. This is a nonlinear problem which can not be solved for directly; however, it can usually be solved for with an iterative least squares approach. To determine pose of a target from triangulated results, the positions of at least four non-coplanar feature points are required to be able to compute both position and orientation without ambiguity.

The added constraint of knowing the location of feature points in the MRB coordinate frame make it possible for a monocular system to determine the pose of the MRB, though with lower accuracy than a stereo-vision system. Since triangulation is not possible with a monocular system it can not determine the 3D position of an individual point in the scene. There is one exception, when the feature points are constrained to be in a plane parallel to the image plane and with known sep-

aration between the planes. Advance knowledge of the scene can provide enough information to make a second image unnecessary for determining pose of an MRB. For pose recovery of the MRB, the positions of known feature points on the MRB are combined with the camera model resulting in a set of nonlinear constraint equations that, when solved, yield the pose of the MRB.

There are several monocular motion analysis systems, both as commercial products and as research projects. These single camera systems have the advantage of having smaller computational requirements then stereo systems because they incorporate less hardware, less image processing, and require no correspondence matching between image sets. However, they also suffer from low resolution and potential significant measurement error in the focal axis of the camera.

By capturing a sequence of images at known time intervals with synchronized cameras the concept of motion analysis can be introduced. The motion of the MRB can be analyzed by recovering the pose of the MRB for each set of images yielding a step wise trajectory of the pose. The difference in pose between sequential sets of images and the frame rate can be used to determine the velocity and accelerations terms: essentially the differentiations of the trajectory of pose with respect to time. This motion analysis method does give a history of the pose, velocity and acceleration of the MRB; however, it is prone to noise and its efficiency could be improved upon. A more effective approach to the problem of motion analysis utilizes tracking algorithms.

A more detailed explanation of photogrammetry, the colinear theory and the Direct Linear Transformation (DLT) can be found in Chapter 3 and the accompanying Appendices B and C.

2.3.2 Tracking

Tracking has been described as the recursive process of predicting, comparing, and updating [Bar-Shalom 88]. The method and implementation of the tracking, as with sensor selection, must to be dictated by the problem being addressed. Some of the concerns to be addressed are whether the system is linear or not, the number of variables to be tracked, and will the system follow an expected trajectory or is it considered to be random.

Tracking the full 6 DOF of a model ship along with the velocity and acceleration terms requires that the tracking algorithm must be:

- able to handle the nonlinearities associated with collecting data with a camera, a nonlinear device,
- able to follow a continuous unplanned trajectory,
- scalable to match the amount of available data,
- able to integrate data from multiple sensors,
- able to maintain track of a moving target in a noisy environment,
- stable in the presence of expected noise from both the sensors collecting data and the system under observation,
- able to track multiple states simultaneously, namely the pose of the MRB along with the velocities and accelerations of pose, $6 \times 3 = 18$ state variables.

The extended Kalman filter (EKF) was selected as the tracking algorithm for the Argo Project because it met all of the above requirements as well as having been demonstrated to function for tasks similar to that of tracking a model ship with video cameras. The EKF, a recursive suboptimal state estimator for nonlinear systems, is an extension of the popular tracking algorithm, the Kalman filter (KF). The KF and the EKF are both popular and well studied methods, are considered reliable, and have been implemented successfully in a variety of applications ranging from radar tracking of aeroplanes and missiles, machine-vision tracking, and analysing and tracking financial trends in the market [Bar-Shalom 88 & 93, Bozic 94, Faugeras 93, Grewal 93, Wilson 93, Wu 88 & Zhang 92].

To track the pose of an MRB with cameras, it is not necessary to triangulate the position of markers. Feature points in the scene, the image coordinates of the markers and the known position of the markers in the coordinate frame of the MRB provide sufficient data. The triangulation process is simply a data reduction technique that renders image coordinates into identifiable locations in 3-space. Since the goal is not to know the position of each marker attached to the target but rath-

er the pose of the target, it is not necessary to go through the intermediate step of triangulation which may only increase the computation time. A set of nonlinear constraint equations that describe the pose of the target can be formed from: (*i*) the colinear equations that describe the numerical camera model, (*ii*) the known target geometry, (*iii*) the general form of the coordinate transformation between the world reference frame, and (*iv*) the model ship's reference frame and the image coordinates.

The six pose variables in the nonlinear constraint equations can be solved in a variety of ways. They can be solved for with an iterative least squares solution method or they can be built into a tracking algorithm such as the Extended Kalman Filter (EKF). The EKF can also be implemented to track the pose of the target based on the triangulated positions of the markers attached to the mobile rigid body. A detailed discussion of using the EKF for tracking the trajectory of a moving target based on image coordinates of known feature points or triangulated feature point location for both monocular and stereo imaging systems can be found in [Faugeras 93 & Zhang 92]. This method has been successfully demonstrated in two separate research projects that both used the EKF and a known target geometry for a monocular system [Wang 91 & 92, Wilson 93, & Wu 88]. Both of these systems are limited to a small working volume so that small changes in position along the focal axis have a noticeable effect. This same methodology is utilized for tracking the pose of a mobile rigid body (MRB) in this project, but it is extended to include multiple cameras viewing the scene to increase resolution and working volume.

There are tracking algorithms and methods other than the EKF that can track the pose of an MRB based image data captured with video cameras. An iterative least-squares approach can accurately determine the pose of objects in a scene by either fitting triangulated feature points with the known geometry of the target or using image coordinates and target geometry as constraints in a set of colinear equations that include pose terms. This method may be more accurate than the EKF approach and does not require time for the filter to establish a track, but it does not actually

track the target but rather builds a sequence of pose data for each set of image frames. Also, it returns only the target pose from which velocity and acceleration terms must be obtained through differentiation of the pose sequence. This is the approach taken by [Sullivan 93 & 97] in his work to develop a system to track the motion of a model ship. Curve fitting techniques could be applied to the pose sequence to fill in the gaps between frames and act as a low pass filter by removing some high frequency noise in the trajectory. Some work has been done to provide a fast noniterative least squares solution to this problem [Weng 92].

Some other methods of following the trajectory of an MRB through video camera captured data are neural networks [Zha 95], a modified Kalman filter approach that assumes the motion follows an affine transformation [Manku], a frequency domain approach using the Fourier transformation [Lin 86], various statistical approaches such as modelling the motion as a Markov process [Aggarwal 97], and model based tracking that is an iterative least squares minimization approach [Aggarwal 97 & Lowe 92]. Some of these methods, although highly accurate, require large computer resources and are very difficult to implement in real-time applications at a reasonable cost. At the other extreme are very simplistic tracking methods that are easy to implement but are not very reliable. In such cases, if the signal is noisy and/or if the target moves somewhat erratically, it may loose track of the target too easily.

The Kalman filter theory and that of the EKF, along with its implementation in the Argo Project, is presented in Chapter 4 of this thesis.

2.4 Related Research Projects

Industry, university and government researchers working on motion capture-analysis problems typically concentrate their efforts in the areas of motion analysis and image processing. They tend to rely on commercial off the shelf (COTS) cameras and computer interfaces as they are well developed and are widely available. On going research and development by industry is done to improve existing commercial systems in order to stay abreast of advancements in technology and

remain competitive. Researchers are working on improving human and animal motion tracking for medical, sports and entertainment-related activities. Others are working on improving rigid body motion tracking for automated inspection and for controlling and evaluating industrial, military, and exploration robots.

Of the researchers working on rigid body motion, a few are working on the problem of tracking the motion of model ships in towing tank and manoeuvring basin environments [Alexander 97, Gospodnetic 92, Sullivan 93 & 97, & Veillon 96]. Another area of common research is visual servoing of an end effector of a robot manipulator arm. This is being explored with both a monocular camera system and the Kalman filter [Wang 92 & Wilson 93] and a stereo camera system coupled with a learning algorithm as the tracker [Zha 95]. Others have been more general in their approach, and are working on the generic problem of tracking rigid body motion from images [Lowe 92, Miller 80, Shapiro 77, Wu 88], camera calibration [Yuan 89], photogrammetry [Krishnan 92] and the combination of all three [El-Hakim 92].

The reverse problem of resolving observer's motion by object tracking is of interest for photography-based mapping from a moving platform, controlling autonomous robots, and the movie industry. An autonomous robot requires knowledge of where it is so that measurements made by the robot can be identified and to aid in navigation for following a preplanned path or to know if the destination has been achieved. The position of the observer can be determined from images of landmarks in the scene with both known and unknown locations [Borenstein 96, Shan 95, Yuan 89]. This is in effect an on-the-fly calibration of the cameras exterior orientation. For an autonomous underwater vehicle (AUV) [Hallan 83] explores the problem of determining the observer's position from sonar images. The source of the images is not relevant as the basic approach for tracking based on image data is common to a variety of image types. In the movie industry, it is necessary to know the observer, namely the camera, pose for integrating computer-generated animation with live characters and scenes.

Additional work is being done by researchers considering the extraction of motion from a sequence of full images using the concept of optical flow [Faugeras 93, Horn 86, Nesi 96 & Sezan 93]. Optical flow is based on the relationship between the temporal variation of image intensity and motion. This approach has some advantages over a token tracking system: it requires no modification of the object to be studied since no target need be attached, it can be used to study non rigid structures, and it can analyse multiple sources of motion in different parts of the scene. However, this method has an extremely high computational cost. Thus, this method is difficult to implement in real-time without expensive specialized computational hardware. Furthermore, solving the correspondence between stereo images is a considerable challenge using this method. As technology advances and our collective understanding of artificial vision increase this will likely be the direction pursued by many as it will give a better evaluation of the motion in the scene.

The reader is referred to table A.1 in Appendix A, for a summary of some of the motion analysis systems under development in a research environment, reviewed as part of this research project. These projects have, for the most part, been demonstrated to function at varying levels of completeness and are typically oriented to a specific task.

2.5 Summary

The task of tracking a manoeuvring model ship is a multifaceted problem. It requires the appropriate sensors, data reduction techniques, and data assimilation techniques. For the Argo Project a multi-video-camera systems was selected as the sensor system for the primary reason that it does not interfere or constrain the model ship in any way. Secondary reasons are that this system will provide sufficient accuracy, resolution, range and sample rate for the desired experiments and data analysis. The Qualisys, Mac Reflex™ hardware was selected for the Argo Project. This hardware consists of one to seven custom video cameras, each with a synchronized IR strobe and a video processor that controls the camera and performs preliminary data reduction that extracts potential feature points and transfers them to the host computer for analysis. The data assimilation

and analysis algorithm combines the Direct Linear Transformation (DLT) camera model for further data reduction and the Extended Kalman Filter (EKF) to assimilate and combine data from different data streams as well as estimating the trajectory of the pose of the mobile rigid body (MRB) along with the velocity and acceleration profiles for that trajectory.

Chapter 3: Photogrammetry

3.1 Introduction

Photogrammetry is the technique of making reliable measurements through the use of photography. With the invention of the camera in the early part of the 19th century, photogrammetry soon followed. Aimé Laussedat, the father of photogrammetry, was the first to develop and use photogrammetric techniques to produce perspective drawings of the facade of l'Hotel des Invalides in Paris, 1849. He carried on to develop new camera equipment and to develop topographical mapping based on photography [Blachut 89].

Traditional photogrammetry is done with specialized film cameras, and stereo plotters, an opto-electro-mechanical device used to locate targets in stereo images so that its position in space could be computed. Presently there is a transition from older analogue equipment and methods to newer digital technology, resulting in hybrids of the two systems. As the cost of high resolution digital-images comes down and digital storage techniques become more efficient, the eventual transition to fully digital photogrammetry will ensue.

With the establishment of photogrammetry to extract quantitative data about the scene from images, the next logical step was to analyse motion with images. Moving film and video cameras capture a sequence of still images separated by a fixed known time interval. By replaying this sequence the motion in the scene is in effect captured. By applying photogrammetric techniques to the individual images and considering the time interval separating the images; time based quantitative data such as changing positions, velocities, accelerations can be extracted. Provided that the frame rate is sufficiently fast to fully observe the motion, this technique can be used to study the motion of almost anything.

A great deal of work has been done in this area in the study of human and animal motion to improve sports skills, evaluate medical treatments, and generate human motion for animation used

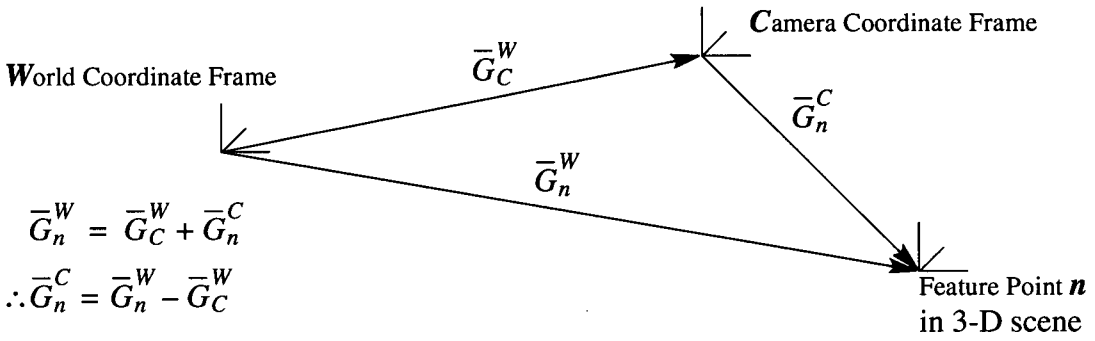
in video games, movie special effects, animated movies and cartoons. The majority of motion capture systems commercially available are designed to develop trajectories of individual points, needed for human motion studies, but not the trajectory of the pose of a MRB with a few exceptions as noted in the previous chapter.

An image is a two dimensional (2D) representation of a three dimensional (3D) scene. As a result, depth information about the scene is lost, thus 3D information can not be extracted from a single image without additional information. This information can be obtained from another image with a different point of view of the same scene, or from *a priori* knowledge of the 3D scene that can be used to apply additional constraints to the image.

Human beings have an apparent depth perception when examining a photograph. They use an existing knowledge base of the environment that the image represents and apply a set of rules. For example, trees on the edge of a lake: in the image they appear to grow out of the water but we know that this is wrong. We know that they are growing on the shore adjacent to the lake so we infer from the picture that they are growing on the shore, although the image does not show this.

3.2 Coordinate Frame Definitions

A simple interpretation of photogrammetry is that it is the combination of two geometric transformations; a coordinate frame transformation of a point from the World coordinate system to the camera coordinate frame followed by a plane perspective projection of the points in the viewer centred coordinate frame onto the image plane of the camera. Figure 3.1 depicts the fundamental coordinate frames necessary for the following descriptions and explanations of how an image is generated in the camera and how to obtain useful information from that image. In the following sections a generic camera coordinate frame will be referred to by C, later when it becomes necessary to differentiate between camera coordinate frames the variable m will be used.



where:

$\bar{G}_n^C = (x_n^C, y_n^C, z_n^C)$ = coordinates of feature point n in camera C

$\bar{G}_n^W = (x_n^W, y_n^W, z_n^W)$ = coordinates of feature point n in the World coordinate frame.

$\bar{G}_C^W = (x_C^W, y_C^W, z_C^W)$ = position of focal point (camera frame C) in the World coordinate frame.

Figure 3.1: Feature point position in the Camera's and the World coordinate frames

3.3 The Pin Hole Camera and Colinear Theory

The simplest camera model used in photogrammetry is the pin hole camera, illustrated in Figure 3.2. The pin hole model is a simple perspective projection onto an image plane that is usually orthogonal to the focal axis. A ray of light emanating from an object point, in the scene, passes through the focus point and then intersects the inverted image plane, yielding an image point. These three points coincide on the same line, hence they are colinear. This is the basis of the colinear theory. It is important to note that the focal point of the camera and the origin of the camera's coordinate frame are coincident.

A feature point in the scene (FP_n) or world space has a unique mapping to a point (IP_n) in the image plane. However, the converse is not true, a point in the image plane does not have a unique mapping to a point in the world space. Due to this fact, a single camera can be calibrated from a set of known points in the scene, but the image coordinates from a single camera cannot give a unique solution for the location of a point in the scene. This unique mapping is described using the following plane perspective projection equation pair that provide image coordinates from feature points in a camera centred coordinate system, Equation (3.1) [Gosine 96 & Weng 92a]:

$$u_n - u_p = \frac{x_n^C \cdot f}{z_n^C} \quad , \tag{3.1.a}$$

$$v_n - v_p = \frac{y_n^C \cdot f}{z_n^C} \quad , \tag{3.1.b}$$

where:

- (u_n , v_n) = image coordinates of feature point n,
- (u_p , v_p) = image coordinates of principal point,
- f = focal length,
- (x_n^C, y_n^C, z_n^C) = position of feature point n in the camera coordinate system C.

To be able to view a scene that is in a coordinate system other than that of the camera a transfor-

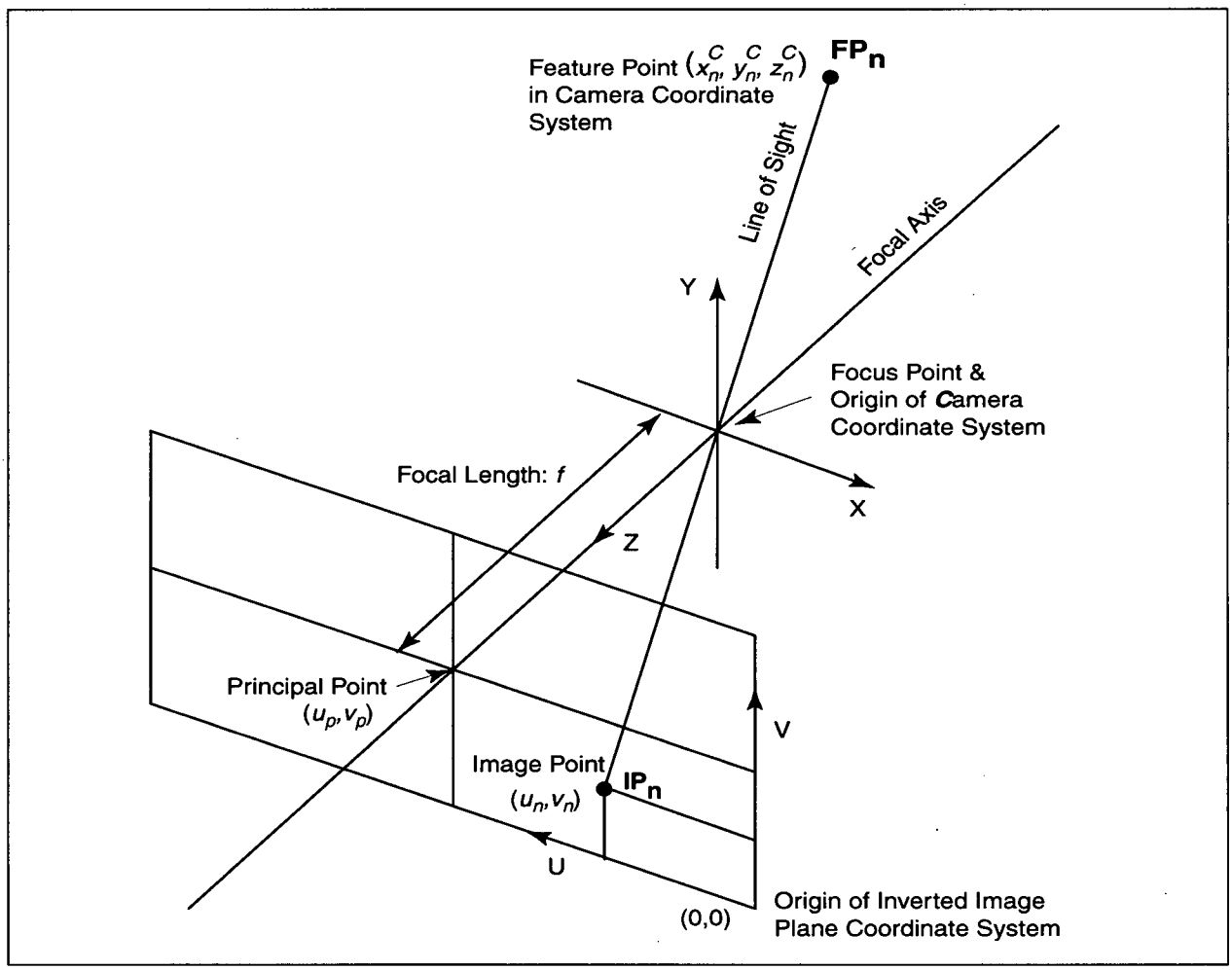


Figure 3.2: Pin Hole Camera

mation is required. A feature point in the scene in the world coordinate system is transformed into the camera coordinate system so that it can be projected onto the image plane. This transformation is based on the position and orientation, the pose, of the camera in the world coordinate system. Combining this transformation and Equation (3.1) yields a new camera model based in the world coordinate system, Equation (3.2) [Aziz 74].

$$u_n = u_p - f_u \cdot \left(\frac{R_{1,1}(x_n^W - x_C^W) + R_{1,2}(y_n^W - y_C^W) + R_{1,3}(z_n^W - z_C^W)}{R_{3,1}(x_n^W - x_C^W) + R_{3,2}(y_n^W - y_C^W) + R_{3,3}(z_n^W - z_C^W)} \right) , \tag{3.2.a}$$

$$v_n = v_p - f_v \cdot \left(\frac{R_{2,1}(x_n^W - x_C^W) + R_{2,2}(y_n^W - y_C^W) + R_{2,3}(z_n^W - z_C^W)}{R_{3,1}(x_n^W - x_C^W) + R_{3,2}(y_n^W - y_C^W) + R_{3,3}(z_n^W - z_C^W)} \right) , \tag{3.2.b}$$

where:

- (f_u, f_v) = focal lengths in horizontal and vertical components of image plane (typically $f_u = f_v$),
- R = a [3 x 3] orthogonal rotation matrix based on $(\psi_C^W, \theta_C^W, \phi_C^W)$,
- $(\psi_C^W, \theta_C^W, \phi_C^W)$ = orientation of the *C*amera coordinate system in the *W*orld coordinate system,
- (x_C^W, y_C^W, z_C^W) = position of focal point (camera frame C) in the *W*orld coordinate frame,
- (x_n^W, y_n^W, z_n^W) = coordinates of feature point *n* in the *W*orld coordinate frame.

To evaluate the image coordinates (u_n, v_n) in Equation (3.2) for the corresponding feature point (x_n^W, y_n^W, z_n^W) an assumption is made that the remaining parameters are known. These parameters are broken into two groups: interior orientation or intrinsic parameters and exterior orientation or extrinsic parameters. The interior orientations are: principal point location (the point where the focal axis intersects the image plane), focal length, pixel size, and systematic error correction terms for image plane distortions and lens distortions. The exterior orientations are the position of the focal point and the orientation of the focal axis and the image plane, which define the reference frame of the camera, with respect to the world coordinate frame. This is the definition of the pose

of a camera. These yet unknown terms are found through calibration.

Calibration of a camera is no different than that of any other sensor system. It is the search for a transfer function relating known inputs to measured outputs. For a camera, the known inputs are the locations of feature points in the scene in the world coordinate system that the camera is to be calibrated against. The measured outputs are the image coordinates corresponding to the feature points in the scene. Solving for the unknown values in the general form of the non-linear camera model transfer function, Equation (3.2), is typically done with an iterative least squares method specific to the camera model being used.

Certain experimental and industrial applications require the camera(s) to move to cover a large area. This is important to note for the application of studying the motion of model ships, where the cameras may be mounted on a moving carriage, to follow the model ship, along the length of a towing tank. In the case of a mobile camera system, the reference frame in which the camera(s) were calibrated, can move but the interior and exterior orientations must be fixed relative to this reference frame. Additional data regarding the movement of the mobile camera base reference frame would have to be included into the tracking module to recover pose, velocity and acceleration terms with respect to a fixed frame of reference. This is not considered in this work.

3.4 Systematic Errors

Colinear theory represents an ideal pin hole camera; however, in reality a camera deviates from this model. This deviation from theory is a combination of systematic errors: image deformation and lens distortion. Image distortion is related to physical deformation and limitations of the media used to capture the image, film or photosensitive electronics. For film it relates to unflatness of the film during exposure and elastic or plastic deformation of the film resulting from winding, changes in temperature, or handling during processing and developing. For electronic image capture systems, such as a CCD array, image distortion can be related to variations in sampling rate and synchronization, the physical alignment of the imaging surface due to manufacturing defect and

vibration, thermal stresses, and the physical limitations of the sensor saturation (over exposure) and quantization limitations (under exposure). Image distortion can be modelled with a combination of two dimensional affine transformations.

Lens distortion can be divided into two components: radial and decentering. Radial distortion is the bending of the ray that connects the feature point and the focus point towards the focal axis to intersect the image point. The term "decentering" refers to the lens focal axis not intersecting the image plane at the principal point. The decentering distortion initially results from a manufacturing defect but can vary with time with handling and the environment that the lens is used and stored in.

Lens distortions can be modelled with polynomials. These distortions were not considered in the current implementation of this project because it is a proof of concept prototype that will be refined in future versions. Detailed explanations of image and lens distortions can be found in [Gosine 96, Karara 89, Tsai 85 & Weng 92a].

Systematic error correction terms (du_n, dv_n) can be added to the camera model to improve its representation of a real camera and in turn its accuracy, Equation (3.3) [Aziz 74]. These terms are typically computed in the calibration process, through the iterative least squares methods. The implementation of this project in its current phase did not include systematic error; however it could be computed as:

$$u_n + du_n = u_p - f_u \cdot \left(\frac{R_{1,1}(x_n^W - x_C^W) + R_{1,2}(y_n^W - y_C^W) + R_{1,3}(z_n^W - z_C^W)}{R_{3,1}(x_n^W - x_C^W) + R_{3,2}(y_n^W - y_C^W) + R_{3,3}(z_n^W - z_C^W)} \right) , \tag{3.3.a}$$

$$v_n + dv_n = v_p - f_v \cdot \left(\frac{R_{2,1}(x_n^W - x_C^W) + R_{2,2}(y_n^W - y_C^W) + R_{2,3}(z_n^W - z_C^W)}{R_{3,1}(x_n^W - x_C^W) + R_{3,2}(y_n^W - y_C^W) + R_{3,3}(z_n^W - z_C^W)} \right) , \tag{3.3.b}$$

where:

- (u_n , v_n) = image coordinates of feature point n,

- (u_p , v_p) = image coordinates of principal point,
- du_n , dv_n = systematic error correction terms for image coordinates (u_n , v_n),
- (f_u , f_v) = focal lengths in horizontal and vertical components of image plane (typically $f_u = f_v$),
- R = a [3 x 3] orthogonal rotation matrix based on $(\psi_C^W, \theta_C^W, \phi_C^W)$,
- $(\psi_C^W, \theta_C^W, \phi_C^W)$ = orientation of the *C*amera coordinate system in the *W*orld coordinate system,
- (x_C^W, y_C^W, z_C^W) = position of focal point (camera frame C) in the *W*orld coordinate frame,
- (x_n^W , y_n^W , z_n^W) = coordinates of feature point n in the *W*orld coordinate frame.

3.5 Direct Linear Transformation (DLT)

The Direct Linear Transformation (DLT) is one implementation of the colinear theory, the pin hole camera model, Equation (3.3) [Aziz 74, Karara 89, Marzan 76, Miller 80 & Shapiro 77]. The DLT was originally designed to compute feature point locations in a scene from a pair of stereo pictures, taken with non-metric cameras. The image coordinates of the feature points are measured and then the location of each feature point is computed using the DLT triangulation equations. The triangulation procedure requires image coordinates from at least two images taken from different points of view. Image coordinates from additional points of view can be included into the triangulation, making the problem over-determined and the error can be minimized, using a least squares solution. This data reduction system is based on feature points in the image and not the image as a whole, on a camera by camera basis. Appendix C provides a derivation of the DLT equations, as well as the calibration and triangulation equations.

Through algebraic manipulation of Equation (3.3) the basic DLT equations are [Aziz 74]:

$$u_n + du_n = \frac{L_1 x_n^W + L_2 y_n^W + L_3 z_n^W + L_4}{L_9 x_n^W + L_{10} y_n^W + L_{11} z_n^W + 1} \quad , \tag{3.4.a}$$

$$v_n + dv_n = \frac{L_5 x_n^W + L_6 y_n^W + L_7 z_n^W + L_8}{L_9 x_n^W + L_{10} y_n^W + L_{11} z_n^W + 1} \quad , \tag{3.4.b}$$

where:

- (u_n, v_n) = image coordinates of feature point n,
- (x_n^W, y_n^W, z_n^W) = feature point n in scene in *W*orld coordinate frame,
- $L_{1\text{-}11}$ = DLT parameters,
- (du_n, dv_n) = systematic error correction terms for feature point n.

The DLT data reduction method has several advantages:

- it is based on feature points and not the whole image,
- it is independent of the means used to obtain the images,
- it can be used with off the shelf non-metric cameras,
- it can be used with any number of cameras (1 to N),
- it has a straight forward implementation.

These advantages, combined with consultation with other researchers working with similar technology in an unrelated field, resulted in its selection for this project.

The eleven DLT parameters, $L_{1\text{-}11}$ in Equation (3.4) [Aziz 74], can be calibrated for with a direct method if the systematic errors are ignored, because they are assumed to be small and negligible, and if the locations of the calibration feature points are accurately known. An iterative least squares method must be employed for a full calibration that includes systematic error correction terms (du_n, dv_n) that are included in the camera model to improve accuracy. In the current implementation of the Argo Project, the direct method is used; however, systematic error correction will have to be included in future versions as error can be significant.

3.6 Photogrammetry and Pose Extraction

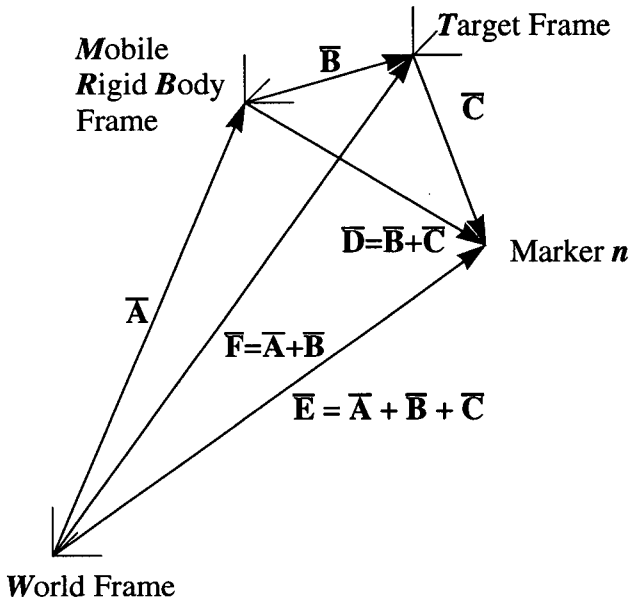
By using two images taken from different points of view, the 3-D location of a point in the scene can be inferred by triangulation. Triangulation is the projection of two rays from the image points through their respective focal points and out until they intersect. The point at which the two rays intersect is the location of the feature point in the scene. One systematic error associated with

triangulation arises when the projected rays do not intersect. To deal with this problem, a best guess approach can be used. If additional data is available from another camera, the triangulation becomes an over-determined problem, and can be resolved by least squares error minimization. Triangulation is a well studied problem and the methods of stereo photogrammetry can be found in a variety of image processing and photogrammetry texts [Duda 73, Faugeras 93, Gosine 96, Horn 86 & Nalwa 93]. A significant problem in triangulation is locating corresponding feature points between two images to the feature point in the scene, namely the correspondence problem. This problem is dealt with in the next section.

Pose extraction from a set of images of a mobile rigid body (MRB) requires prior knowledge of its geometry relating to identifiable and observable features on the body. For example an observer can determine both the position and orientation of a coffee cup from the location of the handle. However, one has difficulty determining a full description of the orientation of a drinking glass as it is symmetric and has no identifiable feature.

In this project, the feature points in the scene are retroreflective spherical markers. Through a combination of special illumination, filters, high contrast black and white images and preliminary image processing the features are reduced to points that represent the centroids of the markers. The markers appear as light spots on a dark back ground, and firm-ware in the image processing units identifies them and computes their centroids along with their height and width dimensions. A spot usually corresponds to a marker but this is not guaranteed. As a result only simple pose extraction techniques can be employed, thus eliminating pose determination from shape and edge detection and other more elaborate methods [Faugeras 93 & Horn 86].

To work within the constraints of having only image centroids of markers as input and to achieve the requirements that the MRB have identifiable and observable features, a target is attached to it. The pose of a MRB with respect to the **W**orld frame, described by vector $\overline{\mathbf{A}}$ in Figure 3.3, can be inferred from the pose of the target in the world frame, vector $\overline{\mathbf{F}}$ in Figure 3.3, provided



$\overline{A}$ = pose vector of ***MRB*** in ***W***orld frame, unknown.

$\overline{B}$ = pose of ***T***arget in ***MRB*** frame, known from target placement on rigid body.

$\overline{C}$ = position of marker n in target, known from target geometry.

$\overline{D}$ = position of marker n in ***MRB*** frame, known from $\overline{B} + \overline{C}$.

$\overline{E}$ = position of marker n in ***W***orld frame, measured value from triangulation.

$\overline{F}$ = pose of ***T***arget in ***W***orld frame.

Figure 3.3: Pose of Mobile Rigid Body and Target in World frame.

that the target's pose relative to the MRB's coordinate system, vector $\overline{B}$ in Figure 3.3, must be known. The target is comprised of markers with known locations in the target coordinate system. To avoid ambiguities the target must have a minimum of four non-coplanar markers to be able to determine its orientation and position, that is, its pose. The pose of an object with respect to a given frame of reference is mathematically represented, as a transformation between two coordinate systems.

The determination of pose of a target comprised of markers with known fixed locations from a set of images of a target can be done in a variety of ways. With a multi-camera systems the position of each marker on the target can be triangulated. Once the positions of the markers making up the target have been triangulated, two different least squares methods can be employed to obtain the pose of the MRB that the target is attached to. The first method solves for the transformation from the MRB coordinate system to the world coordinate system directly. The pose is then obtained by decomposing the resulting transformation to its six basic variables. The second method uses least squares to solve directly for the six variables that describe pose from 3·N nonlinear equations,

where N refers to the number of markers in the target. See Appendix B for details on both methods.

A single camera system also has the ability to measure the pose of a MRB, with reduced accuracy due to low resolution in the focal axis of the camera. By using the dispersion pattern of rays projected through the image points, and the focal point out into the test volume of the scene, along with the geometry of the target, the pose can be determined. By forcing each marker position to lie on the appropriate ray a set of constraints can be developed and the pose of the target is determined and in turn the pose of the MRB.

A variation of this method is to combine the pose transformation of the MRB, to the world frame, and a camera model into a pair of constraint equations, for each marker visible in the image. [Wang 91] Combining the roll-pitch-yaw-translation transformation and the DLT camera model results in Equation (3.5). The derivation of this pair is located in Appendix E.

$$u_n^m = \frac{L_1^m \cdot x_n^W + L_2^m \cdot y_n^W + L_3^m \cdot z_n^W + L_4^m}{L_9^m \cdot x_n^W + L_{10}^m \cdot y_n^W + L_{11}^m \cdot z_n^W + 1} \quad , \tag{3.5.a}$$

$$v_n^m = \frac{L_5^m \cdot x_n^W + L_6^m \cdot y_n^W + L_7^m \cdot z_n^W + L_8^m}{L_9^m \cdot x_n^W + L_{10}^m \cdot y_n^W + L_{11}^m \cdot z_n^W + 1} \quad , \tag{3.5.b}$$

where:

$$\begin{aligned} x_n^W &= \cos(\phi_{MRB}^W)\cos(\theta_{MRB}^W)x_n^{MRB} \\ &+ (-\sin(\phi_{MRB}^W)\cos(\psi_{MRB}^W) + \cos(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\sin(\psi_{MOB}^W))y_n^{MRB} \\ &+ (\sin(\phi_{MRB}^W)\sin(\psi_{MRB}^W) + \cos(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\cos(\psi_{MOB}^W))z_n^{MRB} \\ &+ x_{MRB}^W \end{aligned} \quad , \tag{3.5.c}$$

$$\begin{aligned} y_n^W &= \sin(\phi_{MRB}^W)\cos(\theta_{MRB}^W)x_n^{MRB} \\ &+ (\cos(\phi_{MRB}^W)\cos(\psi_{MRB}^W) + \sin(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\sin(\psi_{MRB}^W))y_n^{MRB} \\ &+ (-\cos(\phi_{MRB}^W)\sin(\psi_{MRB}^W) + \sin(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\cos(\psi_{MRB}^W))z_n^{MRB} \\ &+ y_{MRB}^W \end{aligned} \quad , \tag{3.5.d}$$

$$\begin{aligned} z_n^W &= -\sin(\theta_{MRB}^W)x_n^{MRB} + \cos(\theta_{MRB}^W)\sin(\psi_{MRB}^W)y_n^{MRB} \\ &+ \cos(\theta_{MRB}^W)\cos(\psi_{MRB}^W)z_n^{MRB} + z_{MRB}^W \end{aligned} \tag{3.5.e}$$

This method has the advantage that it requires only four non-coplanar markers to be visible in

the image, and any additional visible markers are used to minimize the error in pose estimate. If a marker is not visible for any reason, the system of constraints can still be used to estimate the pose. An additional benefit is that data collected with any additional cameras can easily be added to the set of constraint equations. With the previously mentioned multi-camera methods, a marker had to be visible in at least two images so that its position in the scene could be triangulated. With this method if a marker is only visible in one image it will still be used to provide two constraint equations. This method yields up to 2·M·N nonlinear constraint equations with six unknowns, M is the number of cameras used and N is the number of markers used. The six unknown pose terms may be solved for using a least squares method.

A multi-camera system has a greater computational cost because it has to build a correspondence map for the feature points for each image, and produces more constraint equations which must be solved. The advantage of this method is that it increases resolution and measurement volume, while lowering the measurement error.

This combination of a pose transformation and a camera model is the basis of the observation model implemented in the extend Kalman filter pose trajectory tracker, discussed in chapter 5. The extended Kalman filter (EKF) is presented in chapter 4.

3.7 The Correspondence Problem

The correspondence problem is matching the image of a feature with the corresponding feature in the scene. This is a significant problem and must be addressed before information about the scene can be recovered. Stereo triangulation relies on an image point in image A that has at most one unique matching image point in image B. The correspondence problem is an ambiguous one as depicted in Figure 3.4. As shown there are three possible triangulation solutions for the locations of three feature points, feature sets (1, 2, 3), (4, 6, 8) and (5, 7, 9). Once paired, the set of image coordinates need to be identified in order to establish correspondence of target points. Single camera and some multi-camera systems require a correspondence map between points in the

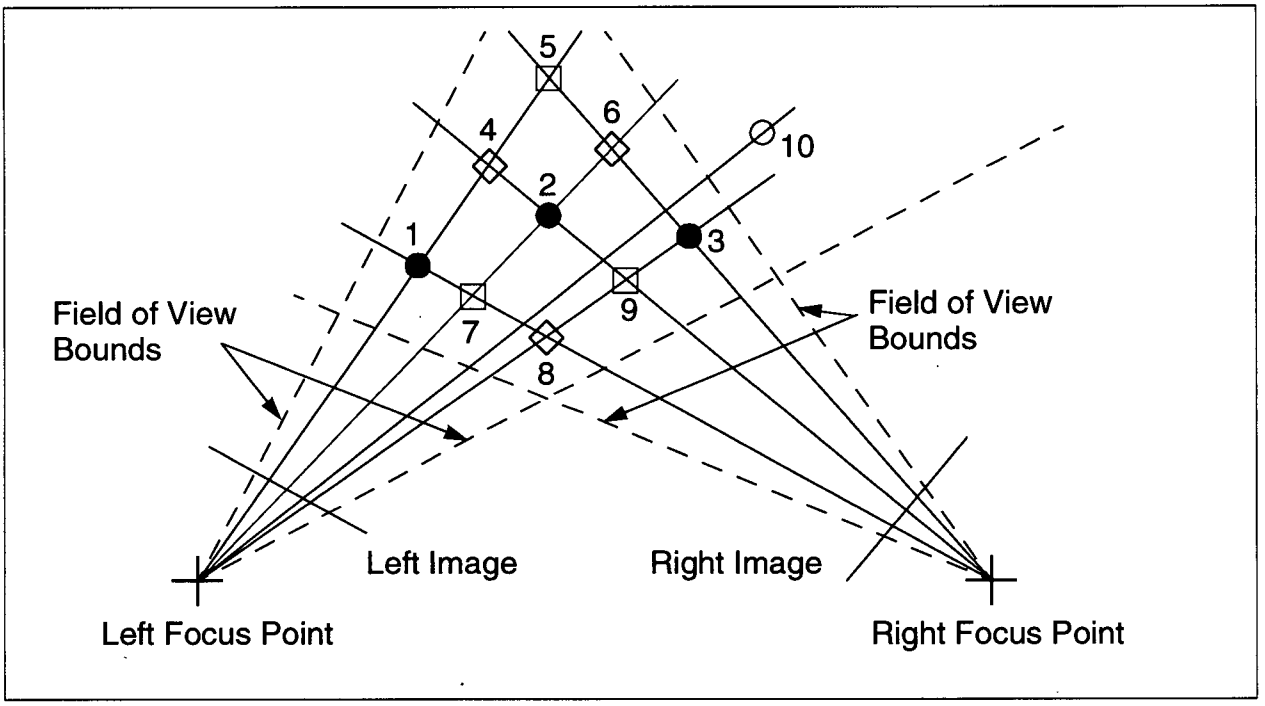


Figure 3.4: Ambiguous Correspondence of Stereo Images [modified form Nalwa 93]

image and the feature point in the scene associated with that image point. The identity of an image point is crucial to obtaining correct and useful information about the scene from a set of images.

Researchers have applied varied solutions to the correspondence problem. This section reviews some of the methods relevant to the implementation of this project. The reader is directed to the literature for further details of the methods presented and descriptions of those not mentioned here, [Faugeras 93, Nalwa 93 & Zhang 92].

Some additional problems that can make the correspondence problem more difficult to solve are: [Zhang 92]

- **Occlusion**: A feature point may be hidden from view from the target interacting with its environment, the fixed scene, another target, or itself: markers can line up along a line of sight making only one marker visible for two. In Figure 3.4, if the feature point set (4,8,9) is considered although all of these features are inside the bounds of the FOV of both cameras feature 4 is not visible in the right image because it is occulded by feature 9, as they lie on a common

line of sight.

- **Disappearance**: Disappearance is when a feature leaves the bounds of the field of view (FOV) of a camera. This prevents the matching of that feature with other images, from another point of view that still have the feature in view. This is definitely a concern if an individual camera can not cover a large test volume with its FOV and maintain sufficient resolution to preform the test. It may be necessary to have multiple cameras, with crossing fields of view to fully cover a large test volume, so as to maintain sufficiently high resolution. If a marker is visible in only one image, there is no correspondence between images and its position can not be triangulated. For example in Figure 3.4, feature 3 is visible in the FOV of both cameras; however, if feature 3 moves to the position of feature 10 then it will leave the FOV of the right camera and will have disappeared from view.
- **Ghosts**: Phantom or ghost markers are points visible in an image but have no corresponding valid feature point. They may be visible from multiple points of view and as a result can have correspondence between them but still they are noise and need to be filtered out before passing data onto the tracking or triangulation procedures. They can be caused by stray reflections off a shiny piece of metal or the surface of a liquid. This a particular problem when doing tests of a model ship in waves. If in Figure 3.4, feature 10 is considered a ghost marker and the feature set (1, 2, 3) are the real features then this will pose no problem when examining the right image but in the left image two problems will arise. The first problem is there are too many features in the left image for the number of expected features; therefore, which are the valid images. The second problem is that if the images of features 3 and 10 appear to overlap in the left image because of close proximity of the lines of sight for these two features then the centroid extraction of the feature could be shifted away from the real centroid of features 3 towards that of feature 10, thus corrupting the measured value taken from the image. Another scenerio that could cause a problem is if feature 10 is still considered a ghost marker and the feature set (1,

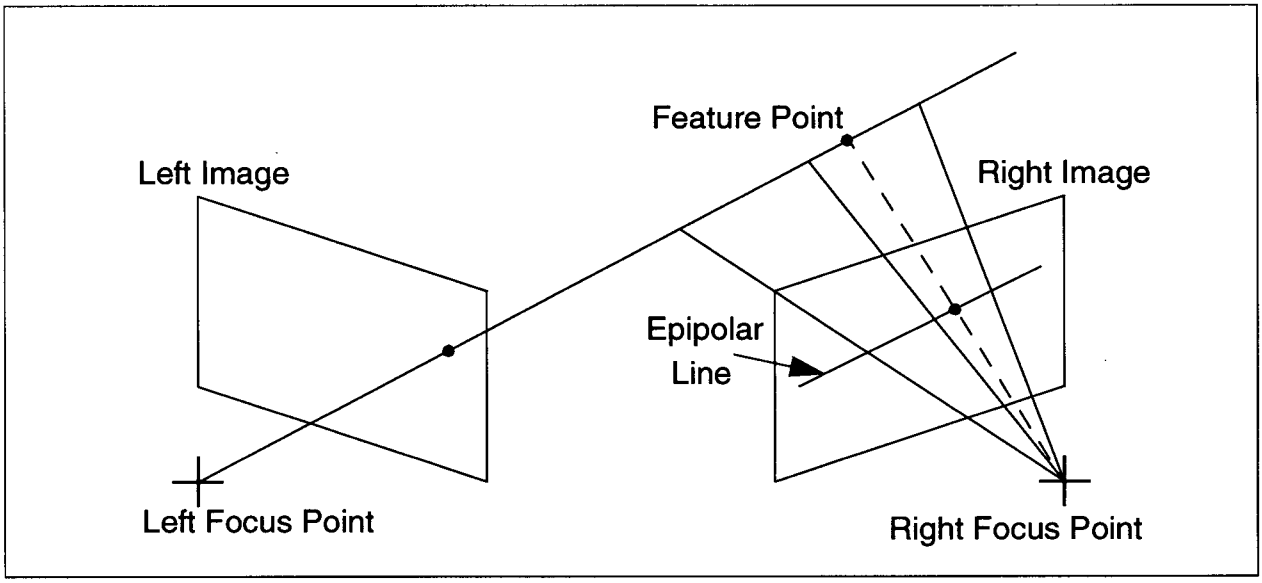


Figure 3.5: Epipolar Constraint [Nalwa 93]

3, 8) are the real features. In this case the position of feature 10 could be taken as the position of feature 3, as feature 3 is occluded by feature 8 resulting in no extra features in the left image. By accepting feature 10 as feature 3 in the left image an error is introduced that not only will cause incorrect results, it may also cause an instability in the tracking algorithm when feature 3 emerges from behind feature 8.

- **Absence**: A feature may not be visible in the FOV that should be. The lack of this feature is not due to occlusion, disappearance, or error in feature extraction. For example, if active markers such as LEDs are being used and one of the LEDs should fail to light for some reason during a test, this would be absence.
- **Appearance**: A feature may move into the FOV that was not previously visible. The new feature may cause confusion with existing features already visible. Appearance can result from an occluded feature becoming visible, an intermittent active marker returns from absence, or a disappeared feature returns to the field of view.

To aid in solving the correspondence problem between two images, a geometric constraint called the epipolar constraint can be used as shown in Figure 3.5. A ray is project out into the scene

through the image point, the focal point and the feature point. This ray is then projected onto the image plane of the second camera, appearing as a line. This line is the epipolar line and the image point in the second camera corresponding to the one in the first image must lie on that line. The epipolar line may be thought of as the intersection of the epipolar plane and the second image plane. The epipolar plane is defined by three points: the focus points for the first and second images and the image point in the first image [Faugeras 93, Nalwa 93 & Zhang 92].

If the range of depth of the feature point in the scene is known and bounded then by projecting these boundaries onto the epipolar line in the second image, this limits the section of the epipolar line in which the image point may be located.

An additional constraint that may sometimes be used depending on the scene and camera placement is monotonic-ordering. Monotonic-ordering is when the left to right order of feature points in one image have the same left to right ordering in the second. This is shown in Figure 3.4 for the correspondence solution of feature set (1, 2, 3). However, this constraint is violated by the other possible solutions, for feature sets (4, 6, 8) and (5, 7, 9).

Some systems take advantage of colours and unique geometric shapes to differentiate markers and aid in the correspondence process. Constraints from lighting and image intensity as well as edge continuity are also used in determining correspondence between images of the features and those in the scene. Hard corners, the intersection of two lines in an image can provide readily identifiable and observable features in a scene. Their geometry allow them to be easily identified in different images, providing correspondence between images and the scene. These cues and others are unfortunately lost and can not be exploited in the Argo project because the imaging system used returns only image coordinate data of suspected marker images.

For a single camera system the correspondence problem involves identifying those feature points in the scene that correspond to the image points. The geometric constraints used to match points in a stereo system can not be applied to a single camera system. The method used in the

Argo project is to make an estimate of the image coordinates with known correspondence and compare the estimates with the incoming image, [Allen 96 & Wang 91]. The incoming marker images are identified by matching them up with their corresponding estimate using a nearest neighbour search.

The estimate of the image could be as simple as the previous image in the sequence or one generated from a dynamics model coupled with and observation model, as implemented in the EKF tracking module. This method has some definite advantages; it removes ghost markers in the image because they have no corresponding estimate. As well, it identifies occluded or disappeared markers when the estimate has no corresponding image, and the estimate-measurement error is scaled down to meet available data. Also, if a valid marker appears in the image by entering the FOV, or emerging from an occluded position, the estimate of this feature will identify it and it will then be included into the list of valid measurements.

3.8 Summary

This chapter contains an overview of the theory of how an image of scene is generated and how to obtain information about the scene that it is observing. The first section of this chapter presented a brief history of photogrammetry and an introduction to the remaining sections. The pin hole camera and colinear theory section describes the function of a pinhole camera and how it can be used to model a real world camera. The systematic errors section explained the deviation from theory observed in real world cameras including lens and image distortions. The Direct Linear Transformation (DLT) is a linearized camera model based on the pin hole camera model. The DLT camera model was selected for the implementation of this project because it was designed for off the shelf non-metric cameras and it is based on feature points rather then the whole images captured by the cameras. This model is well suited to the camera and video processor hardware this project is intended to be used with. Photogrammetry and pose extraction from the images captured is a fundamental component of tracking the pose of a mobile object undergoing MRB motion. Incorporating

the DLT camera model into a tracking algorithm, such as the Kalman filter discussed in the next chapter, allows for the tracking of a target's trajectory based on camera collected data. The final section dealt with the correspondence problem. This significant problem must be dealt with before any information about the scene can be extracted from the image. If the correspondence problem is dealt with incorrectly the interpretation of the image may yield appearantly valid data, yet be incorrect.

Chapter 4: State Estimation, Tracking and the Extended Kalman Filter

4.1 Introduction

As described in Section 1.2, the purpose of the Argo Project is to track of the pose of a scale model ship using multiple video cameras. In general terms, the problem is trajectory tracking of a randomly moving target using data collected with multiple nonlinear sensors in a noisy environment. This type of tracking problem is commonly tackled with an extended Kalman filter (EKF) [Faugeras 93]. To make the text generic to tracking the trajectory of an object based on video images, the scale model ship will be referred to as the mobile rigid body (MRB).

The extended Kalman filter (EKF) was selected as the tracking algorithm for the Argo Project because it can be used to model nonlinear systems, and has been demonstrated to function in similar vision-based token tracking systems [Faugeras 93, Hosie 95 & Wilson 93]. The term token can refer to feature points or markers, lines that define edges of rigid objects, and even 3 dimensional objects such as a cube [Faugeras 93].

In 1960, R. E. Kalman presented a new recursive approach to data filtering based on the Weiner filter [Kalman 60]. The Kalman filter (KF) is a linear recursive filter that generates an optimal estimate of the state of a dynamic system from a model of the system's dynamics and a set of direct and/or indirect measurements of the state variables. The state of the system refers to a set of variables that describe the inherent properties of the system at a particular instance. The discrete Kalman filter is ideally suited to computer implementation because it is linear and it samples data at discrete intervals and models system dynamics as discrete events. The KF has been employed for a variety of tasks: data smoothing, trajectory tracking, forecasting and prediction of future trends, multi-sensor data fusion, and system parameter identification [Bar-Shalom 93]. The flexibility of this mathematical construct has seen it successfully applied in a wide variety of unrelated sectors:

military, finance, medical, industrial, and meteorology [Bar-Shalom 93 & Grewal 93]. As a result of this wide spread use, the KF is well studied and well documented in the literature [Anderson 79, Bar-Shalom 88, Bozic 94, Chui 91, Grewal 93 & Welch 97].

In the early 1960's following Kalman's revolutionary paper, S. F. Schmidt and his team at the National Aeronautics and Space Administration (NASA) [McGee 85] began to explore the possibility of extending the Kalman filter (KF) to the application of nonlinear state estimation problems. As a result of this work, Schmidt introduced the extended Kalman filter (EKF). The EKF linearizes the non-linear state estimation problem by evaluating partial derivatives of the nonlinear state and observation constraint equations at the values of the estimated state variables for each iteration [McGee 85 & Schmidt 70]. Linearization of the problem permits the application of the linear Kalman filter equations to the task of state estimation. In recognition of Schmidt's work, the EKF is also known as the Kalman-Schmidt filter [Grewal 93]. Similar to the KF, the EKF is a well studied technique and is well documented in literature [Bar-Shalom 88 & 93, Chui 91, Grewal 93 & Welch 97].

The EKF is required for the Argo project because of the nonlinear observation model relating pose of the mobile rigid body (MRB) to the image coordinates captured from video images. The basis for the observation model was presented in Chapter 3 along with a more detailed derivation in Appendix E. The EKF has been successfully applied to the problem of tracking the trajectory of a mobile rigid body (MRB) from a sequence of images [Faugeras 93, Hosie 95, Wang 91, Wilson 93, Wu 88].

The large number of authors of literature regarding Kalman filters, and the wide range of topics covered have resulted in variations in terminology and variable names. Some of the notable variations are:

- the terms plant and dynamics model are used interchangeably.
- the terms observation and measurement are used interchangeably, although, by definition,

measurements are observations related to the state of the system.

- $a(s_k, k)$ and $f(s_k, k)$ are the non-linear functions that represent the plant/dynamics model.
- A_k, F_k, and Φ_k are the variable names used for the linear/linearized dynamics matrix models.
- $c(s_k, k)$ and $h(s_k, k)$ are the two variable names used for the non-linear observation function.
- C_k and H_k are the variable names used for the linear/linearized observation matrix models.
- Typically x_k represents the state vector and z_k is used for the measurement vector. To avoid confusion with the position variables x_β^α and z_β^α, the state and measurement vectors are represented hereinby s_k and g_k respectively. This substitution of variable names is also found in literature.

This chapter will introduce the portions of the linear Kalman filter and the extended Kalman filter relevant to the Argo Project. The theory of the linear KF is presented in section 4.2 because it is the basis for the EKF, presented in section 4.3. Section 4.4 discusses design and practical considerations relating to the implementation of the KF and EKF.

4.1.1 State Variables

The state of a system describes the condition or inherent properties of the system being modelled at a particular instance, represented by a set of j state variables. These state variables are the parameters in the state-space equations that numerically model the system. In general, a j^{th} order ordinary differential equation can be transformed into a system of j, 1^{st} order ordinary differential equations by variable substitution and order reduction. The resulting system of equations are the state equations, with j state variables, and the system is then termed to be in state space. These linear equations can then be manipulated with linear algebra. The number of state variables represents the degrees of freedom of the dynamic system being modelled [Grewal 93]. In general, the state variables are collected together into a single j-element state vector, s_k, where the subscript k is the base variable that describes the current discrete interval. For the application of tracking the pose of an MRB, the position and orientation, along with the corresponding velocity and acceleration terms, are the state variables describing the MRB.

4.2 Linear State Estimation and The Discrete Linear Kalman Filter

The Kalman filter (KF) is a linear recursive optimal state estimator that can be configured for filtering, smoothing and predicting the state of a system. By taking into account the previous state, system dynamics, observations of the system, and statistical models of the expected observation noise and perturbations to the trajectory of the state the KF makes an optimal estimate of the current or future state of the system.

The discrete Kalman filter is indexed at regular length intervals making it suitable for implementation on a digital computer. The incoming data is sampled at each interval; therefore, the data sampling rate is built into the filter. The system's dynamics are modelled as discrete events at each interval. The indexed base variable, represented by the subscript k, can be any independent variable provided that it only advances in one direction with constant interval length. In the majority of applications, the base variable is time, but it could be any advancing independent measurable quantity such as distance travelled measured by a car's odometer.

4.2.1 Linear stochastic system model

The state of the system being modelled evolves in time according to the discrete first order recursive difference equation, Equation 4.1 [Bar-Shalom 88].

$$s_k = A_{k-1}s_{k-1} + B_{k-1}u_{k-1} + \omega_{k-1} \quad , \tag{4.1}$$

Here, s_k is the state vector, u_k is the deterministic input vector and ω_k is the dynamics noise vector. The dynamics model, A_k, is a set of state space equations that describes the rate of change of the state of the system. This model is used to generate the current state of the system, s_k, based on the previous state, s_{k-1}. The input matrix, B_k, defines the relationship between the deterministic input variables and the state variables. For the application herein, there is no control input[1], u_k; therefore, the $B_{k-1}u_{k-1}$ term is dropped from Equation 4.1. Background noise that may corrupt the gen-

1. Any further reference to the variable u in this document, unless otherwise specifically stated, is not related to a control input but rather to the horizontal image coordinate, where the subscript would be assigned a numerical value or a variable other than k.

eration of the state and uncertainty in the dynamics model is modelled as an additive, zero-mean, Gaussian white noise vector, ω_k.

The output of a system will be either directly or indirectly related to the state of the system. The relationship that governs the output of the system defined in Equation 4.1 is modelled by Equation 4.2 [Bar-Shalom 88].

$$g_k = C_k s_k + D_k u_k + \xi_k \quad , \tag{4.2}$$

Here, g_k is the output vector of the system, ξ_k is the measurement noise vector, s_k is the state vector and u_k is the deterministic input vector. The relationship between the current state, s_k, and the observable output, g_k, is the observation model[2], C_k. The inverse of the observation model relates measurements taken of the system to the state of system. The direct transmission matrix, D_k, applies deterministic inputs directly to the output of the system. Since, there are no deterministic inputs, u_k, for this application, the $D_k u_k$ term is dropped from Equation 4.2 and no longer referenced herein. Noise in the data collection process[3] and the level of uncertainty in the sensor is modelled as an additive, zero-mean, Gaussian white noise vector, ξ_k.

Dropping the deterministic input terms from Equations 4.1 and 4.2, yields the two models that are used in the following discussion of the linear Kalman filter:

$$s_k = A_{k-1} s_{k-1} + \omega_{k-1} \quad , \tag{4.3}$$

$$g_k = C_k s_k + \xi_k \quad , \tag{4.4}$$

4.2.2 System Noise and Uncertainty

Uncertainty in the two system models, dynamics and observation, is modelled as noise. The observation noise vector, ξ_k, and the dynamics noise vector, ω_k, are assumed to be additive, uncorrelated (defined by Equations 4.5 through 4.7) and zero-mean Gaussian white noise, (defined in Equation 4.8).

2. In some instances in the literature the observation model is referred to as the measurement model.
3. In most literature related to the Kalman filter the observation noise is referred to by the greek variable ν_k, but to avoid confusion with the vertical image coordinate variable v_k, ξ_k is used in place of ν_k.

$$E\langle \xi_j \xi_k^T \rangle = \begin{pmatrix} \sigma_\xi^2 & , j = k \\ 0 & , j \neq k \end{pmatrix} \quad \text{for all } j \text{ and } k \ , \tag{4.5}$$

$$E\langle \omega_j \omega_k^T \rangle = \begin{pmatrix} \sigma_\omega^2 & , j = k \\ 0 & , j \neq k \end{pmatrix} \quad \text{for all } j \text{ and } k \ , \tag{4.6}$$

$$E\langle \xi_j \omega_k^T \rangle = 0 \qquad \text{for all } j \text{ and } k \ , \tag{4.7}$$

$$E\langle \xi_k \rangle = \mathrm{E}\langle \omega_k \rangle = 0 \qquad \text{for all } k \ , \tag{4.8}$$

It is unlikely that a linear state space model exists that perfectly represents a real world system. A real system has imperfections, potential cross coupling effects, and possible nonlinearities that might not be included in the state space model. Physical systems are also susceptible to environmental uncertainties. Examples of environmental noise are the effect turbulence has on a flying aircraft, and the effect that chop has on the motion of a ship. In both cases the environmental effects have a noticeable local effect, but on the scale of the overall trajectory of the craft these effects can be modelled as small scale, high frequency noise.

Such factors listed above will cause a deviation between the linear dynamics model of how the states interrelate and how they actually interrelate in the real world. The expected effect of these factors is modelled as noise, ω_k, and added to the model to better reflect the real world situation. The level of noise must be estimated by the modeller and may have to be adjusted if its magnitute is too large or too small, to reflect the real world. Unfortunately, the magnitude of the noise at any given interval is not a measurable quantity, and the expected value of the noise, $E\langle \omega_k \rangle$, is zero, as defined in Equation 4.8. Althought the expected value of the noise is zero, the variance of this noise is an assignable value that describes the probable range of the magnitude of the noise. The definition of the covariance of the dynamics noise vector, ω_k, is:

$$Q_k = E\langle (\omega_k - E\langle \omega_k \rangle)(\omega_k - E\langle \omega_k \rangle)^T \rangle \ , \tag{4.9}$$

Substituting Equation 4.8 into Equation 4.9 reduces the dynamics covariance to:

$$Q_k = E\langle \omega_k \omega_k^T \rangle \ , \tag{4.10}$$

Observations made of the system output also contain some uncertainty which is modelled as noise, ξ_k, and included into the observable output model, Equation 4.4. The uncertainty in this process is the combination of short comings in the observation model, C_k, and the physical limitations and characteristics of the sensors and data acquisition equipment used to measure the system output. Potential sources of inaccuracy in the observation model, C_k, are vibrations resulting from not perfectly stiff members, possibly producing unforeseen cross-coupling terms. Another potential source of inaccuracy is imperfect manufacturing of the system that may yield a differing geometry from that expected in the model. Data collection in the observation of the system outputs is subject to its own imperfections. Electronic instrumentation is prone to electronic interference that corrupts the signal, and small nonlinearities in the sensor that can change with time and/or temperature. Care must be taken when estimating the signal to noise ratio as this will effect the confidence in the observed outputs of the system. Since it is not possible to evaluate the actual magnitude of the noise, ξ_k, at each interval, the expected value of this noise, $E\langle\xi_k\rangle$, is equal to zero, as defined in Equation 4.8. As with the dyanmics noise, the vairance of the observation noise is quantifiable. The covariance of the observation noise vector is defined as:

$$R_k = E\langle(\xi_k - E\langle\xi_k\rangle)(\xi_k - E\langle\xi_k\rangle)^T\rangle \quad , \tag{4.11}$$

Substituting Equation 4.8 into Equation 4.11 reduces the observation covariance to:

$$R_k = E\langle\xi_k\xi_k^T\rangle \quad , \tag{4.12}$$

4.2.3 State Estimation

To evaluate the state, s_k, and the corresponding output, g_k, of a system using Equations 4.3 and 4.4 it is necessary to know the uncertainty terms, ω_k and ξ_k, at each interval k. However, as indicated above, this is not possible. Therefore, the best that can be achieved is to estimate[4] the state, $\hat{s}_k$, and output, $\hat{g}_k$, using the expected value of the noise terms. Also, since the state propagation equation is recursive, the initial state, s_0, for k = 1, is required to start the process. With no way to

4. The hat symbol, ^, above a variable name indicates that it is an estimate of the real value.

evaluate accurately the initial state the best that can be done is to use an estimated value, $\hat{s}_0$. Applying this argument to Equations 4.3 and 4.4 yields Equations 4.13 and 4.14.

$$\hat{s}_k = A_k \hat{s}_{k-1} + E\langle \omega_k \rangle \quad , \tag{4.13}$$

$$\hat{g}_k = C_k \hat{s}_k + E\langle \xi_k \rangle \quad , \tag{4.14}$$

Substituting equation 4.8 into equations 4.13 and 4.14 reduces them to:

$$\hat{s}_k = A_k \hat{s}_{k-1} \quad , \tag{4.15}$$

$$\hat{g}_k = C_k \hat{s}_k \quad , \tag{4.16}$$

An evaluation of the effectiveness of the state estimation process described in Equation 4.15 may be done by comparing the estimated output, $\hat{g}_k$, generated by Equation 4.16, with the actual output, g_k, of the physical system.

$$e_k = g_k - \hat{g}_k \quad , \tag{4.17}$$

The Kalman filter (KF) uses knowledge of the expected uncertainty in the estimation process as well as the error between the estimated and measured outputs of the system, which indicate the overall signal to noise ratio at the current interval, to generate an optimal state estimate. The KF operates in a predict-correct loop sequence. The estimate of the current or future state of the system is determined by first making an initial guess as to what the current or future state of the system will be. A correction factor, based on observation of the system and the statistics associated with the initial guess and the observation processes, is added to the initial state estimate to produce a final estimate of the current or future state. As depicted in Figure 4.1, the discrete Kalman filter loop starts with the preliminary estimate of the current state[5], $\hat{s}_k^{\dagger}$, based on the previous state estimate, $\hat{s}_{k-1}$, with the linear Equation 4.15, where A_k represents the set of linear state space equations that make up the dynamics model.

$$\hat{s}_k^{\dagger} = A_k \hat{s}_{k-1} \quad , \tag{4.18}$$

The preliminary state estimate, $\hat{s}_k^{\dagger}$, is then used to estimate of the output values, $\hat{g}_k^{\dagger}$, using the

5. The superscript dagger, †, indicates that this variable is a preliminary estimate only.

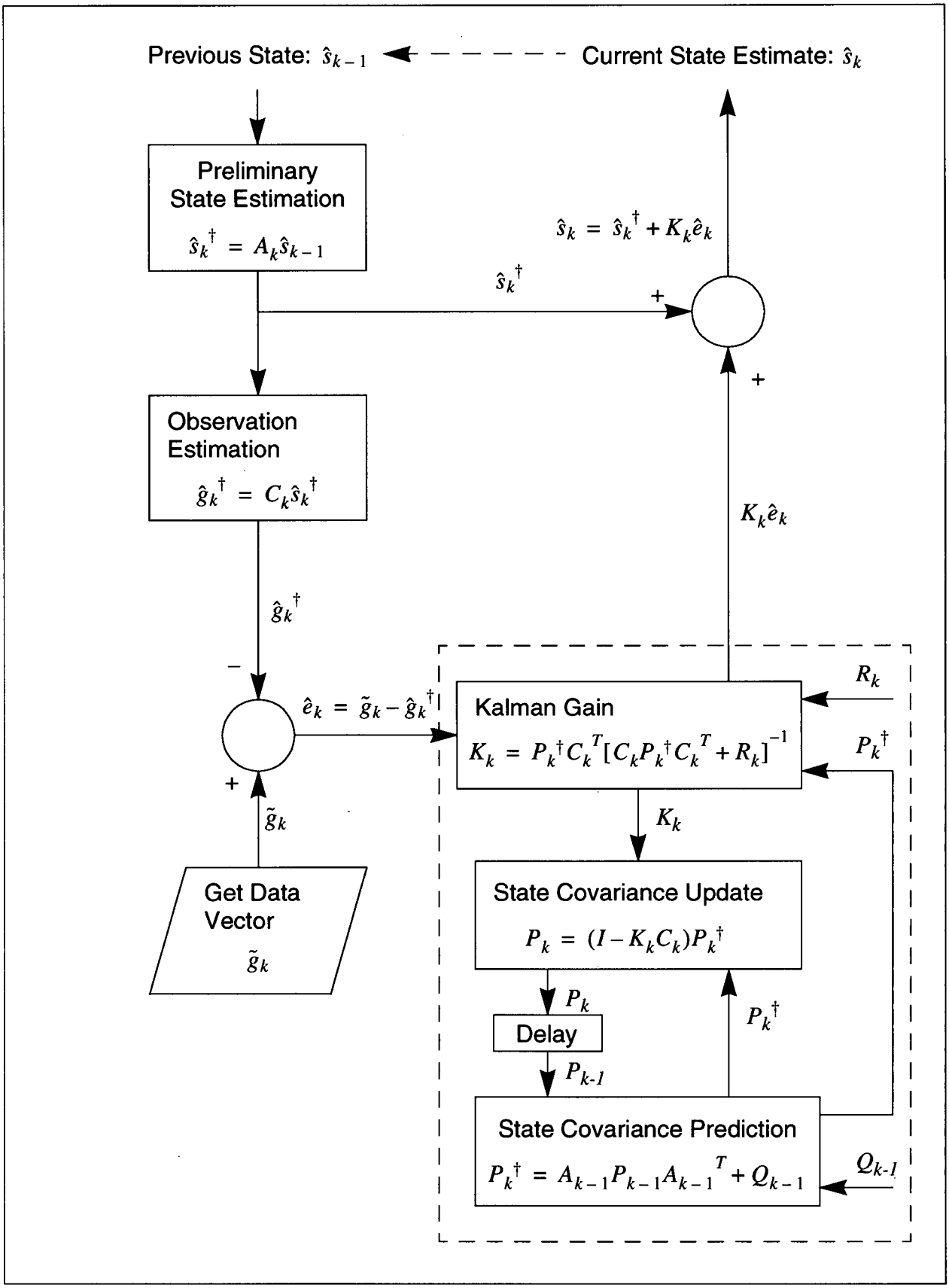


Figure 4.1: Discrete Kalman Filter Flowchart.

observation model, C_k, that relates the state of the system to observations made of the system, equation 4.19.

$$\hat{g}_k^{\dagger} = C_k \hat{s}_k^{\dagger} \quad , \tag{4.19}$$

The estimated error, $\hat{e}_k$, is the difference between values measured from observations made of the system[6], $\tilde{g}_k$, and those predicted from the preliminary state estimate, $\hat{g}_k^{\dagger}$, Equation 4.20:

$$\hat{e}_k = \tilde{g}_k - \hat{g}_k^{\dagger} = \tilde{g}_k - C_k \hat{s}_k^{\dagger} \quad , \tag{4.20}$$

The optimal estimate of the current state, $\hat{s}_k$, is computed by adding to the preliminary state estimate, $\hat{s}_k^{\dagger}$, a correction factor, namely, the estimated error, $\hat{e}_k$, scaled by the Kalman gain matrix, K_k, as shown in Equation 4.21 [Bozic 94].

$$\hat{s}_k = \hat{s}_k^{\dagger} + K_k(\tilde{g}_k - C_k \hat{s}_k^{\dagger}) \quad , \tag{4.21}$$

This recursive process repeats itself for each discrete interval.

4.2.4 The Kalman Gain and the State Error Covariance

The Kalman gain matrix, K_k, scales the error between observed and estimated outputs, $\hat{e}_k$ from Equation 4.20, to minimize the mean square of the state-estimation error. The state-estimation error, e_k^*, is the difference between the actual state, s_k, and the state estimate, $\hat{s}_k$, generated from the KF, as shown in Equation 4.22 [Welch 97].

$$e_k^* = s_k - \hat{s}_k \quad , \tag{4.22}$$

The definition of the covariance of the state-estimation error is shown in Equation 4.23.

$$P_k = E\langle (e_k^* - E\langle e_k^* \rangle)(e_k^* - E\langle e_k^* \rangle)^T \rangle \quad , \tag{4.23}$$

Since the KF minimizes the state-estimation error, the desired and expected value of the error is zero.

$$E\langle e_k^* \rangle = 0 \quad , \tag{4.24}$$

Using Equations 4.22, 4.23 and 4.24, the covariance of the state-estimation error, otherwise known

6. The tilde symbol, ~, over the output variable g indicates that it is the vector of measured values that may be corrupted by the data collection process.

as either the updated, or *a posteriori*, state error covariance, reduces to Equation 4.25 [Welch 97].

$$P_k = E\langle (s_k - \hat{s}_k)(s_k - \hat{s}_k)^T \rangle \quad , \tag{4.25}$$

Another important and similarly defined covariance term that is indicative of the accuracy of the preliminary state estimate, $\hat{s}_k^{\dagger}$, is the predicted, or *a priori*, state error covariance, shown in Equation 4.26 [Welch 97].

$$P_k^{\dagger} = E\langle (s_k - \hat{s}_k^{\dagger})(s_k - \hat{s}_k^{\dagger})^T \rangle \quad , \tag{4.26}$$

These two covariance terms indicate how accurate the filter currently is and how it should be shifted if necessary to minimize the state estimation error. Their influence on the overall estimation process of the KF is achieved by controlling the Kalman gain.

The Kalman gain, K_k, represents the relative confidence between the state estimation and state observation processes and how this relationship effects the overall state error. The confidence of the state estimation and the state observation processes are defined as the dynamics noise covariance, Q_k, and the observaiton noise covarinace, R_k, respectively. Based on these two covariance models the Kalman gain, K_k, is adjusted as the state error covarinace, P_k, varies. A detailed explanation and derivation of the Kalman gain, K_k, can be found in many texts related to the KF [Bar-Shalom 88 & 93, Bozic 94 & Grewal 93]. Inside the dashed box in Figure 4.1, a small recursive loop that self tunes the Kalman gain exists within the greater KF loop, which is itself recursive. To compute the Kalman gain, K_k, the predicted, or *a priori*, state error covariance, $P_k^{\dagger}$, is used in Equation 4.27 [Bozic 94].

$$K_k = P_k^{\dagger} C_k^T [C_k P_k^{\dagger} C_k^T + R_k]^{-1} \quad , \tag{4.27}$$

The calculation of the *a priori* state error covariance, $P_k^{\dagger}$, is expressed in terms of the updated, or *a posteriori*, state error covariance, P_k, as in Equation 4.28 [Bozic 94].

$$P_k^{\dagger} = A_{k-1} P_{k-1} A_{k-1}^T + Q_{k-1} \quad , \tag{4.28}$$

Closing the loop is the calculation for the *a posteriori* state error covariance, P_k, by referring to the Kalman gain, K_k, Equation 4.29 [Bozic 94].

$$P_k = [I - K_k C_k] P_k^{\dagger} \quad , \tag{4.29}$$

This recursion process allows the filter to retain information about past state estimates and regulates the KF by adjusting the confidence level placed on observations and initial state estimates from the dynamics model. In effect, the state error covariance matrix, P_k, acts as the memory of the filter, recording how past events influenced the filter's operation and the confidence levels of estimation and observation. The error covariance tends to increase as noise or instability in the dynamics of the system increase. Changes in the relative confidence levels between the estimation and observation processes is reflected in the Kalman gain, K_k.

An increasing Kalman gain, K_k, indicates a reduction in the confidence level of the preliminary state estimate, $\hat{s}_k^{\dagger}$, by placing higher confidence in the measured values that describe the system. A large gain will tend to make the KF have a rapid response to changes in measured values. Correspondingly, a small gain indicates a high level of confidence in the state estimation process and little correction is required and/or there is low confidence in the measured values recorded. A small gain will make the KF sluggish and slow to respond to changes that are observed, as it will tend to keep with the *a priori* state estimate [Bar-Shalom 93]. Because of this, the Kalman gain should not be too small as to not respond to changes in trajectory and cause the filter to diverge, conversely it should not be too large as it will follow spurious noise in the trajectory that can lead to instability.

4.2.5 Kalman filter regulation

If the error, $\hat{e}_k$, between the initial estimate of the outputs, $\hat{g}_k^{\dagger}$, and the observed values, $\tilde{g}_k$, increases, the confidence in the initial state estimate, $\hat{s}_k^{\dagger}$, decreases, provided that the confidence in the measured values remains constant. This results in an increase in the state covariance, P_k, resulting in an increase in the Kalman gain, K_k. This has the effect of increasing the correction factor added to the initial state estimate, $\hat{s}_k^{\dagger}$, to compute the final state estimate, $\hat{s}_k$.

If the expected variance of the dynamics noise is lower than the actual variance of the distur-

bances, then the magnitude of the dynamics covariance matrix, Q_k, will be too small. If this is the case, then confidence in the preliminary state estimate, $\hat{s}_k^{\dagger}$, will be too high, forcing the state estimation correction, based on observations, to be smaller than necessary for an optimal state estimate. This has the effect of making the filter less responsive to changes in the trajectory of the target that were not anticipated by the design of the dynamics model. If the anticipated signal to noise ratio of the measurement of system outputs is actually too low, then the observation covariance model, R_k, will be too small. In this case, the confidence in the measured outputs, $\tilde{g}_k$, is too high. This has the effect of making the preliminary state estimate, $\hat{s}_k^{\dagger}$, not significant enough in the overall estimate of the current state, $\hat{s}_k$. When the confidence in observations of the system is too high then the filter will tend to follow spurious trajectories resulting from noise artifacts from the data collection process. If R_k and Q_k are both too big then all disturbances are accepted as being within parameter specifications resulting in a sluggish system that lags changes in trajectory. If R_k and Q_k are both too small then the confidence in the models are too high and the filter will follow spurious noise and potentially be unstable.

Since K_k is proportional to the ratio of Q_k/R_k, if system noise levels increase, then the confidence in the dynamics model, A_k, is lessened, implying an increase in the confidence of observations made. This change in confidence is manifest by increasing K_k which has the effect of increasing the correction factor to the preliminary state estimate, $\hat{s}_k^{\dagger}$. If the measurement noise increases then K_k decreases and the confidence in the observations, $\tilde{g}_k$, decreases and more weight is given to the state estimate. If it is possible to monitor the noise levels, the observation and dynamics covariance models can be adjusted up or down, tuning the filter to match the expected noise levels and produce the optimal trajectory estimate.

Because the ability of the KF, used as a tracker, to follow a changing trajectory is sensitive to the expected noise levels, it is necessary select the values of these noise with great care. Another important factor governing the success of the KF to generate an optimal state estimate are the initial

conditions used to start the tracking process.

4.2.6 Kalman Filter Initialization

Special care must be taken when initializing the KF, otherwise it may fail or at the very least take several more iterations than expected to establish track of the target. For the first iteration of the KF the index base variable is set to one, $k = 1$. To initialize the KF a starting state, s_0, and an initial state error covariance, P_0, are required. Also required are the dynamics model, A_0, and the initial dynamics and observation noise covariance models, Q_0 and R_1, respectively. The starting state, s_0, should be as close as possible to the actual state of the system being tracked. It can be determined by using a known starting point, from a best guess of the operator, or by taking measurements of the system and running them backwards through the observation model. The initial state error covariance is more difficult to estimate as it depends on the dynamics of the system, the relationship between measured values and the state of the system and the expected sources of noise. A derivation of the initial state error covariance, P_0, used in the implementation of the Argo Project can be found in Appendix D. To generate the three initial covariance models (P_0, Q_0 and R_1) an estimate of the sources of noise and the expected noise levels are necessary. Special care must be taken when estimating noise levels as they control the sensitivity and confidence of the filter to track the trajectory. If the expected noise levels and sources of noise do not change with time, they are considered constant.

4.3 Nonlinear State Estimation and the Extended Kalman Filter (EKF)

In the event that the system dynamics model and/or the observation model are nonlinear, the direct use of the conventional Kalman filter for state estimation is not possible. However if the problem is first linearized, the linear Kalman filter theory and equations can be applied to state estimation of a nonlinear system.

Nonlinear state propagation takes the general form of Equation 4.30. The nonlinear relation-

ship between system output and the current state is represented in its general form by Equation 4.31. Equations 4.30 and 4.31 are analogous to their linear counterpart Equation 4.1 and 4.2.

$$s_k = a(s_{k-1}, u_{k-1}, \omega_{k-1}, k) \quad , \tag{4.30}$$

$$g_k = c(s_k, u_k, \xi_k, k) \quad , \tag{4.31}$$

As before with the linear system, k is the index base variable, s_k is the state vector, u_k is the deterministic input vector which again is unused and will be ignored in the following discussion, ω_k is the dynamics noise vector, g_k is the system output vector, and ξ_k is the observation noise vector. The function a, in Equation 4.30, represents a set of nonlinear state dynamics equations that relate the previous state of the system, s_{k-1}, to current state, s_k. There is one state dynamics equation for each state variable. The function c, in Equation 4.31, represents a set of nonlinear equations that describe the current output of the system, g_k, based on the current state of the system, s_k. There is one equation for each observable output of the system.

Removing the unused deterministic inputs, u_k, and assuming that the dynamics and observation noise terms are linear, additive, white and conform to the rules previously stated in Equations 4.5 through 4.8, Equations 4.30 and 4.31 become.

$$s_k = a(s_{k-1}, k) + \omega_k \quad , \tag{4.32}$$

$$g_k = c(s_k, k) + \xi_k \quad , \tag{4.33}$$

Making the assumption that the noise is linear and additive allows the use of the noise covariance matrices defined earlier in Equations 4.10 and 4.12 to be used in the same manner in the EKF as in the KF.

Comparing Figures 4.1 and 4.2, it is obvious that there are four differences between the structures of the KF and EKF loops. This comparison also shows the remaining structure of the EKF is taken directly from the KF. The first difference is that the preliminary state estimation process is now a nonlinear process, as shown in Equation 4.34.

$$\hat{s}_k^{\dagger} = a(\hat{s}_{k-1}, k) \quad , \tag{4.34}$$

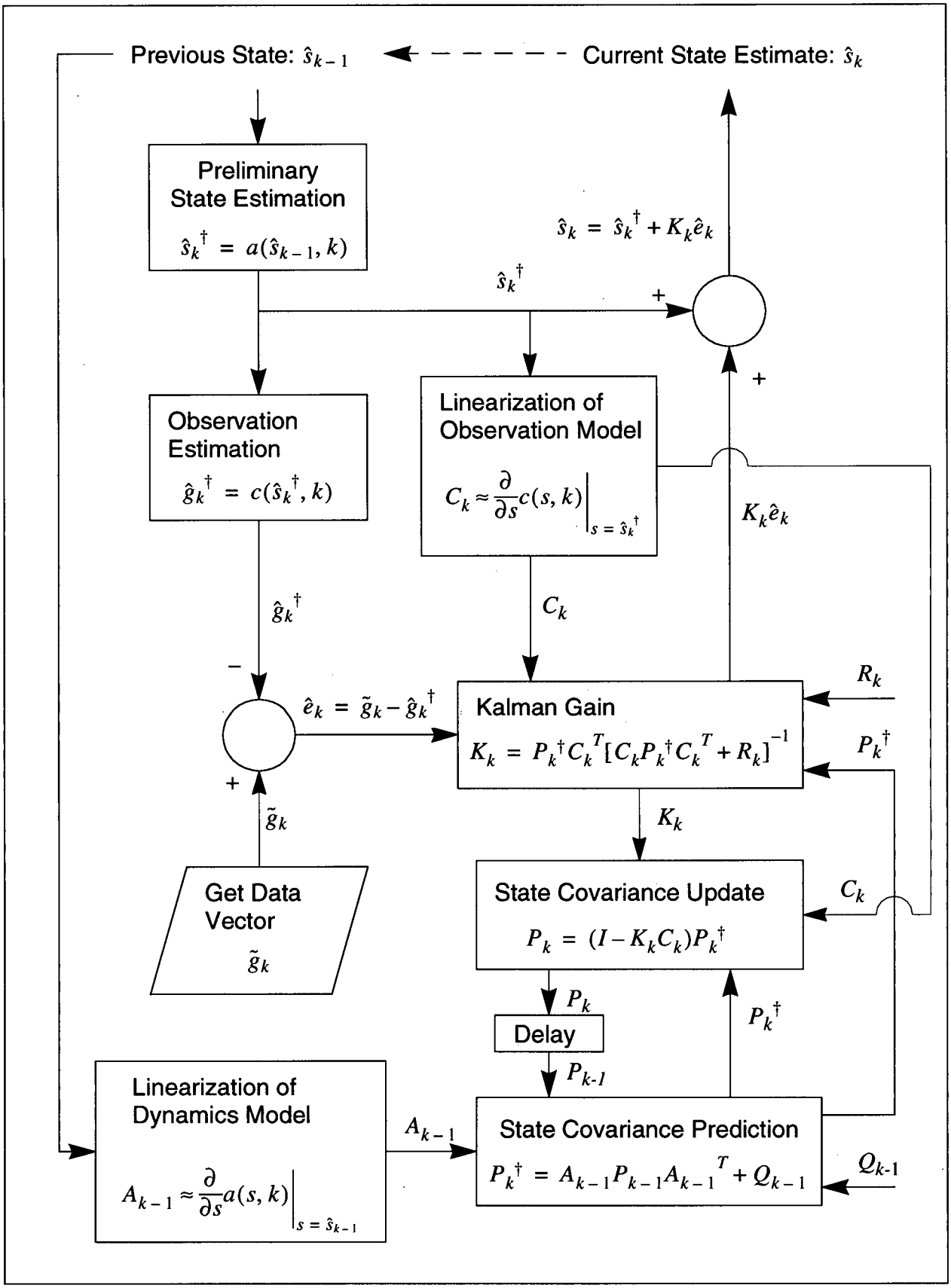


Figure 4.2: Full Discrete Extended Kalman Filter Flowchart.

Secondly, the estimate of the output, $\hat{g}_k^{\dagger}$, based on the preliminary state estimate, $\hat{s}_k^{\dagger}$, is now a nonlinear relationship described by Equation 4.35.

$$\hat{g}_k^{\dagger} = c(\hat{s}_k^{\dagger}, k) \quad , \tag{4.35}$$

The two most notable differences between the KF and the EKF are the addition of the two linearization processes to approximate the dynamics model, A_k, and the observation model, C_k, for computing the state error covariance and the Kalman gain. The justification for these approximations is presented in the following subsection.

Other than these four differences, the EKF functions and behaves in a very similar manner as the conventional KF, described in the previous section. The EKF tends to be less robust than the KF because of errors introduced in the linearization process. This requires that the equations used to model the real system be as accurate as is reasonably possible. Tthe anticipated signal to noise ratios for both the dynamics and observation models must also be carefully selected.

4.3.1 Linearization

Since it not possible to know the actual state of a system, Equations 4.32 and 4.33 can not be evaluated. However, if an estimate of the last state of the system exists, a Taylor series expansion can be used to approximate the function a by evaluating it about the last estimate of the state, $\hat{s}_{k-1}$.

$$a(s_{k-1}, k) \cong a(\hat{s}_{k-1}, k) + \frac{\partial}{\partial s} a(s, k)\Big|_{s = \hat{s}_{k-1}} (s_{k-1} - \hat{s}_{k-1}) + HOT \quad , \tag{4.36}$$

Similarly an approximation of the function c can be done by evaluating the Taylor series about the preliminary estimate of the current state, $\hat{s}_k^{\dagger}$. *HOT* represents the higher order terms which are ignored.

$$c(s_k, k) \cong c(\hat{s}_k^{\dagger}, k) + \frac{\partial}{\partial s} c(s, k)\Big|_{s = \hat{s}_k^{\dagger}} (s_k - \hat{s}_k^{\dagger}) + HOT \quad , \tag{4.37}$$

Substituting Equations 4.36 and 4.37 into Equations 4.32 and 4.33 results in

$$s_k \cong a(\hat{s}_{k-1}, k) + \frac{\partial}{\partial s} a(s, k)\Big|_{s = \hat{s}_{k-1}} (s_{k-1} - \hat{s}_{k-1}) + \omega_k \quad , \tag{4.38}$$

$$g_k \cong c(\hat{s}_k^{\dagger}, k) + \frac{\partial}{\partial s} c(s, k)\Big|_{s = \hat{s}_k^{\dagger}} (s_k - \hat{s}_k^{\dagger}) + \xi_k \quad , \tag{4.39}$$

The derivatives of Equations 4.38 and 4.39, with respect to the state estimates that their Taylor series expansions were evaluated at, are:

$$\frac{\partial s_k}{\partial s}\bigg|_{s = \hat{s}_{k-1}} \cong \frac{\partial}{\partial s} a(s, k)\Big|_{s = \hat{s}_{k-1}} \quad , \tag{4.40}$$

$$\frac{\partial g_k}{\partial s}\bigg|_{s = \hat{s}_k^{\dagger}} \cong \frac{\partial}{\partial s} c(s, k)\Big|_{s = \hat{s}_k^{\dagger}} \quad , \tag{4.41}$$

The linear equivalents to Equations 4.34 and 4.35 are Equations 4.15 and 4.16, and are repeated for the reader's convenience as Equations 4.42 and 4.43.

$$\hat{s}_k = A_k \hat{s}_{k-1} \quad , \tag{4.42}$$

$$\hat{g}_k = C_k \hat{s}_k \quad , \tag{4.43}$$

These two equations have the general form of a line that passes through the origin. In the case of Equation 4.42, the dynamics model, A_k, can be considered as the slope of the line and therefore the derivative of the current state estimate, $\hat{s}_k$, with respect to the previous state estimate, $\hat{s}_{k-1}$. Therefore in the linearization process, it is reasonable to expect that the Jacobian of the vector of nonlinear state equations, $a(\hat{s}_{k-1}, k)$, with respect to the previous estimated state vector, $\hat{s}_{k-1}$, can be assigned as the dynamics model, A_k, as shown in Equations 4.44. The approximated dynamics model, A_k, is then used in the calculation of the state error covarinace prediction, $P_k^{\dagger}$.

$$A_k \approx \frac{\partial}{\partial s} a(s, k)\Big|_{s = \hat{s}_{k-1}} \quad , \tag{4.44}$$

For the output estimation, Equation 4.43, the observation model, C_k, is the slope for the linear relationship between the independent variable, the current state estimate, $\hat{s}_k$, and the estimated output, $\hat{g}_k$. As with the dynamics model it is reasonable to assign that the Jacobian of the vector of nonlinear output equations, $c(\hat{s}_k, k)$, with respect to the current estimated state vector, $\hat{s}_k$, as the slope in the observation model, C_k, shown in Equation 4.45. This approximation of the observation

model is substituted into the linear KF equations in place of the linear version when computing the Kalman gain and the state error covariance update.

$$C_k \approx \frac{\partial}{\partial s} c(s, k)\Big|_{s = \hat{s}_k} , \tag{4.45}$$

4.3.2 About the Extended Kalman Filter

The extended Kalman filter (EKF) is a nonlinear state estimator that takes its general structure from the conventional Kalman filter. The EKF linearizes the nonlinear function(s) about the current estimated state at each discrete interval. The EKF is a sub-optimal filter, unlike the KF which is an optimal state estimator. An optimal nonlinear filter is difficult to realize and is typically computationally intensive [Bar-Shalom 93]. For the sake of practical implementation this accommodation is made and as a result the filter is detuned.

As has been shown for the EKF, the dynamics and observation models are linearized about an estimated state; however, it is possible for a nominal state to be used in place of the estimated state. This process, normally termed as the linearized Kalman filter (LKF), has the advantage that the filter gains and the state error covariance can be computed in advance, off-line of the real-time operation. The disadvantage of this method is that it requires *a priori* knowledge of the state trajectory and it is not very robust against nonlinear approximation errors. If the actual state deviates from this trajectory; then the filter will either fail because of divergence or it will yield unreliable results. For this reason the LKF was rejected for the Argo Project, as it is intended to track the pose of any MRB through any trajectory, without *a priori* knowledge of the trajectory and it should be considered random. The EKF linearizes about an estimated trajectory which leads to more accurate trajectory estimates as it adapts and reacts to perturbations to the trajectory. The main disadvantage of the EKF is that the Kalman gains and the state error covariances can not be precomputed requiring on-line/real-time computations. These are the reasons that the EKF tends to be more robust than the LKF but more difficult to realize in real-time applications

4.3.3 Argo Project EKF requirements

For the Argo Project, the dynamics of the ship motions are modelled with linear kinematics equations, while the observation model that relates camera data and the pose of the MRB is non-linear. This means that the requirements for the state estimation process are for a partial discrete extended Kalman filter where only the observation model is linearized. The EKF loop required for the Argo Project is depicted in Figure 4.3.

4.4 Design and implementation considerations

The KF and the EKF can both fail to track the trajectory of a target for a variety of reasons. The two main modes of failure of the KF and EKF are failure to establish track of the trajectory and failure to maintain track of the trajectory. When the filter fails to maintain track of the trajectory it is said to diverge; this is an all encompassing term that has many possible causes. Failure can be the result of a single problem but is often the reslut of a combination of problems. The EKF tends to be more sensitive than the KF to many of these problems because of the linearization process, making it less stable. Some of the causes for failure of the KF and EKF are listed below [Grewal 93]:

- The (E)KF can diverge if the filter gain, K, becomes small and the observation data are still significant and needed to correct the state estimate.
- The (E)KF uses the dynamics model, A, to approximate how an actual physical system changes state. If this model does not sufficiently represent the physical system, the filter will fail or yield suspect results. An inappropriate dynamics model, A_k, will yield a false preliminary state estimate, $\hat{s}_k^{\dagger}$, that in turn will cause a false estimate of the observable system outputs, $\hat{g}_k^{\dagger}$. It will also cause an error in the state error covariance prediction, $P_k^{\dagger}$.
- The second model used by an (E)KF is the observation model, C_k, that relates measurements made of the dynamic system to the state of the system. A mismatched observation model, C_k,

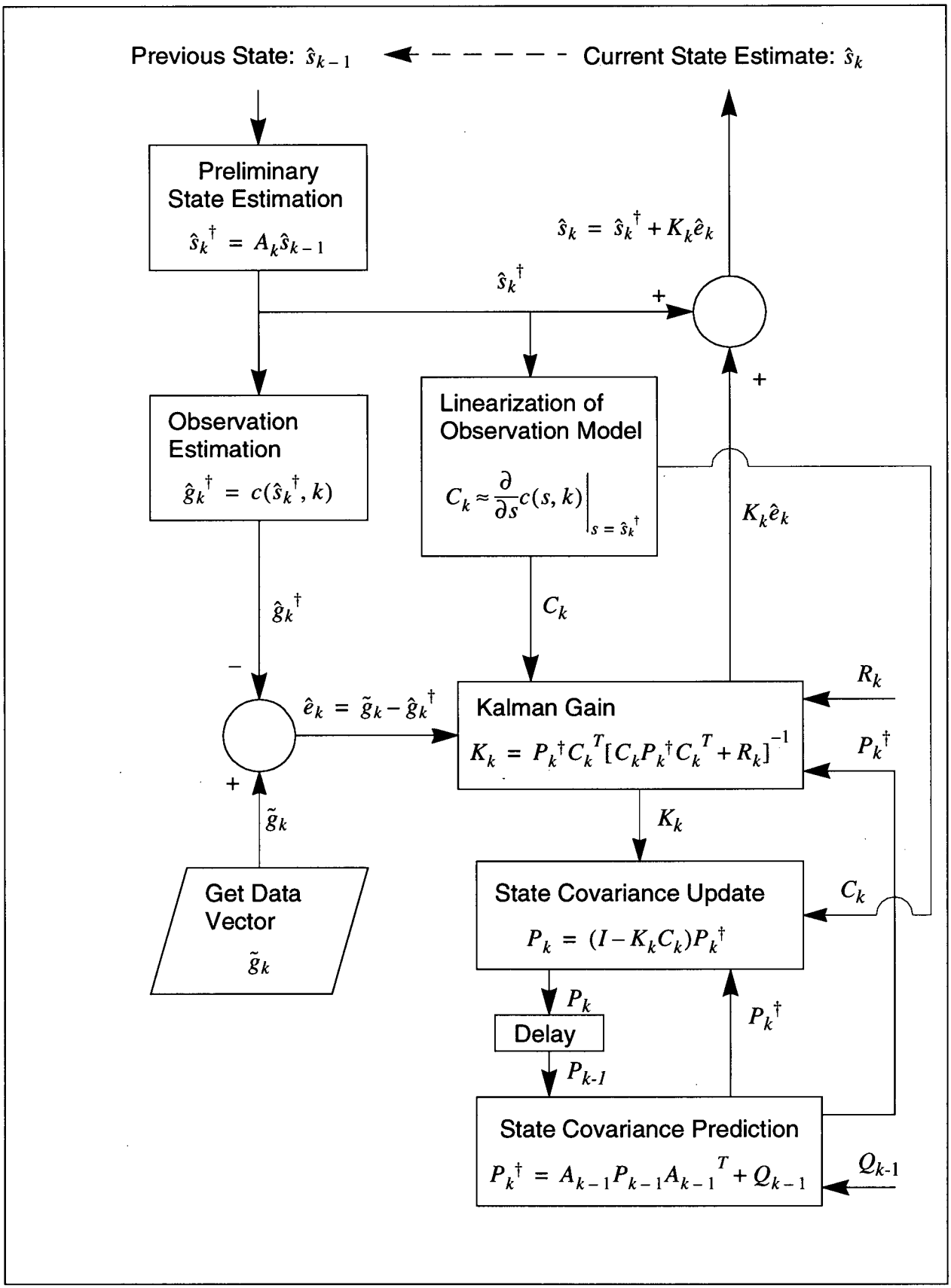


Figure 4.3: Partial Discrete Extended Kalman Filter Flowchart.

will yield a false observation estimate, $\hat{g}_k^{\dagger}$, and will also cause errors in the state error covariance update, P_k, and the Kalman gain, K_k. An error in the observation estimate, $\hat{g}_k^{\dagger}$, will result in an error in the estimated observation error term, $\hat{e}_k$, that will affect the accuracy of the current state estimate, $\hat{s}_k$.

- If the linearized dynamics and observation models do not sufficiently represent the original nonlinear relationships, the EKF will fail. This problem may be overcome by adding higher order terms into the linearization process. However, this should be avoided as this would further deviate the filter from Kalman's original work.
- Improper estimation of the expected noise levels in either the observation and dynamics noise covariance models (Q and R) can lead to either:
  - an unstable filter that will react to noise and spurious trends if the estimated noise levels are too small. Any noise outside the expected bounds will be perceived incorrectly as a change in trajectory.
  - a sluggish filter that lags the actual trajectory if the expected noise levels are too high as small changes in trajectory or measured values can be perceived as allowable noise. Not until the change in trajectory becomes significant will the filter start to adapt to the change and then slowly. In effect the (E)KF acts as a low pass filter: the more sluggish the filter becomes the lower is the cut-off frequency.
- Some ill-conditioned problems relating to implementation of the (E)KF are [Grewal 93]:
  - Errors or large uncertainties in the values of the dynamics, observation and statistical models have not been allowed for in the design of the Kalman filter and may cause it to diverge.
  - Large numerical ranges of values in the state variables, measured values, or the matrix models indicate inappropriate scaling or dimensional units used in the design of the filter. This can lead to round-off errors.
  - The intermediate matrix $[C_k P_k^{\dagger} C_k^{T} + R_k]$ in the Kalman gain calculation could be non-

invertible.

  - Large dynamics and observation models have large matrix dimensions that are computationally and memory intensive to manipulate. The number of arithmetic operations increases by the square or the cube of the matrix dimension, depending on the operation. Each additional arithmetic operation introduces a round-off error and increases the computation time of each iteration through the loop. This can be a limiting factor in real-time systems.
  - Poor machine precision can also result in large round-off errors and lead to numerical instability of the filter.
- The filter can fail to maintain track if the incremental changes between samples are too large. This can be the result of too low a sampling rate or if disturbances are too large.
- False data or falsely interpreted data can lead to confusion and the filter may track an unintended trajectory.
- A good initial estimate of the state of the system is required otherwise the filter may never acquire the track.

The KF and the EKF can be implemented in a parallel structure for parallel processing. This will improve the computational time performance to meet the scheduling requirements of real-time filtering [Chui 91].

4.5 Summary

The Kalman filter is a linear optimal state estimator, ideally suited for tracking trajectories and trends of a dynamic system under either direct or indirect observation. The KF is a very flexible tool and can be implemented to track, to smooth data, to extract a signal from a noisy background and data fusion, and to combine data gathered from multiple sources to observe a system. The Kalman filter can not be directly applied to the problem of tracking a MRB using multiple video cam-

eras because of the nonlinearity of the observation outputs of the system. However, the KF can be applied to the study of a nonlinear system by incorporating a linearization step. The resulting filter is sub-optimal because of the linearization process implemented, and is referred to as the extended Kalman filter (EKF).

Chapter 5: Argo Project Implementation

5.1 Introduction

As stated before, the goal of the Argo Project is to track, in real-time, the rigid body motion of a model ship operating in the simulated sea of a towing tank or manoeuvring basin. The current implementation of this goal is at the proof of concept stage. The computer program can track simulated and experimental data, off-line.

The system has been designed around the Qualisys MacReflex™ hardware so as to be compatible with on-going research programs at NRC-IMD, NRC-Hydraulics, and MUN Faculty of Engineering. With a common hardware base, different projects can concentrate on various functions and resources can be shared.

The research component of this project is in the software that performs image data reduction and tracks the motion of the mobile rigid body (MRB) in the test volume. This research effort involves the combination of video technology, photogrammetry, state estimation and trajectory tracking. The necessary background on these different areas has been presented in the previous three chapters and accompanying appendices. The video technology allows for capturing snap shots in time of a dynamic system under observation. Photogrammetry allows for the interpretation of the pose of the MRB at a particular instance in time. Also, photogrammetry shows the relationship between observable outputs of a system and the corresponding image coordinates recorded, through what is known as a camera model. State estimation and trajectory tracking fuses the data collected from the different video cameras together, introduces the time element, and generates the best estimate of motion of the MRB.

5.2 Methodology

From image coordinates of known feature points of an MRB captured with multiple synchronized video cameras, the Argo Project tracks the six degrees of freedom (6 DOF) of motion of an MRB inside a known test volume. The basis for the tracking algorithm used in the Argo Project is the partially extended Kalman filter (EKF), presented in the previous chapter. This algorithm generates an estimate of the state variables, the pose of the MRB, from observable outputs of the system, the image coordinates corresponding to known feature points on the MRB. A flowchart of the tracking algorithm of the Argo Project is shown in Figure 5.1. It is necessary to use the EKF rather than the regular Kalman filter (KF) because the observable outputs do not have a linear relationship to the state variables that describe the system. This algorithm, along with the associated model, and state variables are described in the following subsections.

5.2.1 State Variables and the System Dynamics Model

For the Argo Project, the state of the system is the pose of the MRB with respect to an inertial world frame of reference, along with the first and second order derivatives of pose with respect to time. For tracking the motion of a model ship, the inertial frame of reference is defined by the geometry of the towing tank or manoeuvring basin. The reference frame of the model ship has its origin at the centre of gravity (CG) of the model ship as defined in Figure 1.1. The six pose terms combined with the associated six velocity terms and six acceleration terms yield 18 state variables, shown in vector form in Equation (5.1).

$$s_k = \begin{bmatrix} \psi^W_{MRB} & \dot{\psi}^W_{MRB} & \ddot{\psi}^W_{MRB} & \theta^W_{MRB} & \dot{\theta}^W_{MRB} & \ddot{\theta}^W_{MRB} & \phi^W_{MRB} & \dot{\phi}^W_{MRB} & \ddot{\phi}^W_{MRB} & \cdots \\ & x^W_{MRB} & \dot{x}^W_{MRB} & \ddot{x}^W_{MRB} & y^W_{MRB} & \dot{y}^W_{MRB} & \ddot{y}^W_{MRB} & z^W_{MRB} & \dot{z}^W_{MRB} & \ddot{z}^W_{MRB} \end{bmatrix}^T_k , \tag{5.1}$$

where:

ψ = roll angle,
θ = pitch angle,
ϕ = yaw angle,
x = surge displacement,
y = sway displacement,
z = heave displacement.

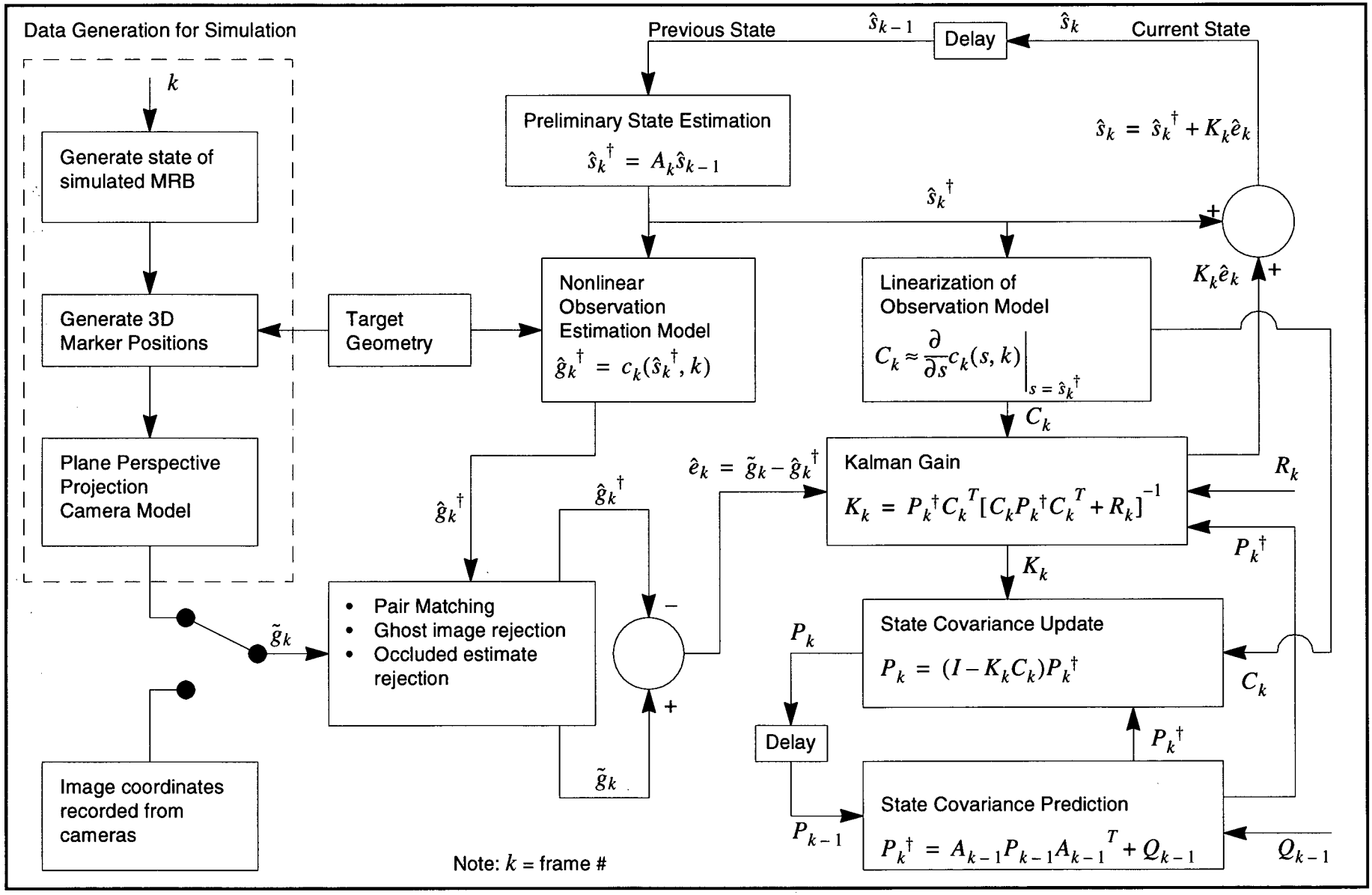


Figure 5.1: Argo Project Tracking Algorithm.

To make the tracking module capable of following any random motion of an MRB, the dynamics model assumes uniformly accelerated linear motion for each degree of freedom with no cross-coupling terms. This simple kinematics model was selected for four reasons: (*i*) it is a linear representation of virtually any motion provided the step-size is small enough, (*ii*) it requires no *a priori* knowledge of the system dynamics and can accommodate a changing system dynamic, (*iii*) it simplifies and minimizes computations as no linearization is required, and (*iv*) it has been demonstrated to function, provided the target motion is sufficiently over-sampled [Wilson 93]. The uniformly accelerated linear motion dynamics model for a given degree of freedom is shown in Equation (5.2).

$$\alpha = \begin{bmatrix} 1 & \tau & \frac{\tau^2}{2} \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix} , \tag{5.2}$$

where:

- τ = sample period of the observations made of the system,
- α = the $[3 \times 3]$ matrix that describes the 3^{rd} order linear motion of a single degree of freedom.

With no cross coupling, the single degree of freedom model is repeated for each degree of freedom in the system dynamics model.

$$A_k = \begin{bmatrix} \alpha & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha \end{bmatrix} , \tag{5.3}$$

Replacing this simplified dynamics model with a more detailed one, specific to the system under observation, would improve the state estimation process. However, this limits the universality of the tracker for use with systems with dynamics that differ from those modelled, and for systems

where the dynamics change during the test period. The situation of a system with changing dynamics is relevant to seakeeping tests of a model ship because of the free surface effects of water on deck or in a compartment will change the system's mass and mass distribution. Furthermore, building a specific dynamics model also requires extensive *a priori* knowledge of the system and how it will perform under all possible conditions that may be experienced during a test.

5.2.2 Observable Outputs and the Observation Model

The observable outputs of the system are measurable physical responses , that either directly or indirectly describe the state of the system. Since the Argo Project is a non-contact measurement system, it is not possible to directly measure the pose of the MRB. Hence, an indirect observable output is necessary. The indirect method chosen for the Argo Project is to use the image coordinates of feature points of known locations on the MRB. The observation model relates the state of the system, the pose of the MRB, to the observable outputs, the image coordinates of feature points on the MRB.

The feature points on the MRB are high contrast markers that make up a target that is rigidly mounted to the MRB. The observable outputs are the image coordinates corresponding to the centroids of these markers. A minimum of four non-coplanar markers are required to define a volume that has a unique pose solution. Therefore, it is necessary that the target attached to the MRB contain a minimum of four non-coplanar markers. Having additional markers in the target is desirable to add a level of redundancy that will potentially lower the overall measurement error and will provide enough data if one or more markers are occluded from view.

Ping-pong-ball-sized spheres were selected as the markers because their centroid has the same apparent position from all angles of view. These spheres are coated with retroreflective paint, giving them a high contrast appearance compared with the MRB and the background of the scene.

Markers are used in place of feature points native to the MRB because they are readily identifiable in an image and they are independent of the MRB. In general, the use of standardized feature

points, independent of the MRB, is advantageous because it is not necessary to reconfigure the feature point extraction process for each new MRB tested. Rather than attaching individual markers directly to the MRB, it is recommended that the markers be attached to the MRB as a collective group, known as a target.

The markers have known positions in the 3D space of the target's coordinate frame. Knowing the pose of the target's coordinate frame relative to that of the MRB's allows for the determination of the marker locations in the MRB coordinate system. This transformation is shown in Equation (5.4). Hence the investigator, studying the motion of an MRB, need only know the pose of the target, rather than having to survey the location of each marker on the MRB for each test schedule. This makes the experimental setup simpler, speedier and more accurate.

$$\begin{bmatrix} x_n^{MRB} \\ y_n^{MRB} \\ z_n^{MRB} \\ 1 \end{bmatrix} = {}_{MRB}\bar{H}_{Target} \cdot \begin{bmatrix} x_n^{T} \\ y_n^{T} \\ z_n^{T} \\ 1 \end{bmatrix}, \tag{5.4}$$

where:

- ${}_{MRB}\bar{H}_{Target}$ is a roll-pitch-yaw-translation homogeneous transformation from the target coordinate frame to the MRB coordinate frame, defined in Section 2 of Appendix B.

The relationship between the state of the system and the observable outputs is known as the observation model. In this project, the observation model is the combination of two separate transformations. The first transformation converts the locations of the target markers from the MRB's coordinate frame to the world coordinate frame, as shown in Equation (5.5). The world coordinate frame is the base reference frame that the pose of the MRB is measured relative to.

$$\begin{bmatrix} x_n^W \\ y_n^W \\ z_n^W \\ 1 \end{bmatrix} = {}_W\bar{H}_{MRB} \cdot \begin{bmatrix} x_n^{MRB} \\ y_n^{MRB} \\ z_n^{MRB} \\ 1 \end{bmatrix} , \tag{5.5}$$

where:

- ${}_W\bar{H}_{MRB}$ is a roll-pitch-yaw-translation homogeneous transformation from the MRB coordinate frame to the world coordinate frame, defined in Section 2 of Appendix B.

The second transformation is a camera model that relates three dimensional (3D) points in the world coordinate system to the two dimensional (2D) coordinates in the image plane. For the Argo Project the Direct Linear Transformation (DLT) was selected as the camera model. Details of the DLT and the reasons for its selection are presented in Chapter 3 and Appendix C of this document. The combination of the DLT and transformation of marker coordinates from the MRB reference frame to the world reference frame results in the nonlinear observation model, represented by $c_k(s_k,k)$ in Figure 5.1. The definition of the observation model is shown in Equations (5.6.a) through (5.6.e).

$$u_n^m = \frac{L_1^m \cdot x_n^W + L_2^m \cdot y_n^W + L_3^m \cdot z_n^W + L_4^m}{L_9^m \cdot x_n^W + L_{10}^m \cdot y_n^W + L_{11}^m \cdot z_n^W + 1} , \tag{5.6.a}$$

$$v_n^m = \frac{L_5^m \cdot x_n^W + L_6^m \cdot y_n^W + L_7^m \cdot z_n^W + L_8^m}{L_9^m \cdot x_n^W + L_{10}^m \cdot y_n^W + L_{11}^m \cdot z_n^W + 1} , \tag{5.6.b}$$

where:

$$\begin{aligned} x_n^W = {} & \cos(\phi_{MRB}^W)\cos(\theta_{MRB}^W)x_n^{MRB} \\ & + (-\sin(\phi_{MRB}^W)\cos(\psi_{MRB}^W) + \cos(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\sin(\psi_{MRB}^W))y_n^{MRB} \\ & + (\sin(\phi_{MRB}^W)\sin(\psi_{MRB}^W) + \cos(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\cos(\psi_{MRB}^W))z_n^{MRB} \\ & + x_{MRB}^W \end{aligned} , \tag{5.6.c}$$

$$y_n^W = \sin(\phi_{MRB}^W)\cos(\theta_{MRB}^W)x_n^{MRB} + (\cos(\phi_{MRB}^W)\cos(\psi_{MOB}^W) + \sin(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\sin(\psi_{MRB}^W))y_n^{MRB} + (-\cos(\phi_{MRB}^W)\sin(\psi_{MRB}^W) + \sin(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\cos(\psi_{MRB}^W))z_n^{MRB} + y_{MRB}^W \quad , \tag{5.6.d}$$

$$z_n^W = -\sin(\theta_{MRB}^W)x_n^{MRB} + \cos(\theta_{MRB}^W)\sin(\psi_{MRB}^W)y_n^{MRB} + \cos(\theta_{MRB}^W)\cos(\psi_{MRB}^W)z_n^{MOB} + z_{MRB}^W \quad , \tag{5.6.e}$$

where:

- $(\psi_{MRB}^W, \theta_{MRB}^W, \phi_{MRB}^W, x_{MRB}^W, y_{MRB}^W, z_{MRB}^W)$ are the state variables that represent the pose of the MRB's coordinate frame relative to the world's coordinate frame.
- (u_n^m, v_n^m) are the image coordinates of marker *n*, in the image plane of camera *m*, a pair of observable outputs.
- L_{1-11}^m are the DLT parameters for camera *m*.

The eleven DLT parameters, L_{1-11}^m, for each camera are determined through calibration. A complete description of the calibration procedure is given in Section C.2.

The nonlinear observation model function, c_k, generates the vector of observable outputs, g_k, using the following loop structure:

for $m = 1$ to M

for $n = 1$ to N

$$g_k[(2 \times N \times (m-1) + 2 \times n - 1), 1] = u_n^m \tag{5.7.a}$$

$$g_k[(2 \times N \times (m-1) + 2 \times n), 1] = v_n^m \tag{5.7.b}$$

end

end

As part of the EKF execution, the nonlinear observation model is linearized by taking the gradient of the observation outputs, g_k, with respect to the state vector, s_k, as shown in Equation (5.8),

$$C_k \cong \frac{\partial}{\partial s} c(s, k)\Big|_{s = s_k} \quad . \tag{5.8}$$

The resulting linearized observation model, C_k, is a matrix that is comprised of alternating rows corresponding to the alternating pattern of the (u, v, u, v) of the observation output vector, g_k. The

general form of the rows that make up the linearized observation model, C_k, are as follows. Row $2 \times N \times (m-1) + 2 \times n - 1$ of the linearized observation matrix is shown in Equation (5.9.a),

$$\left[\frac{\partial u_n^m}{\partial \psi_{MRB}^W} \; 0 \; 0 \; \frac{\partial u_n^m}{\partial \theta_{MRB}^W} \; 0 \; 0 \; \frac{\partial u_n^m}{\partial \phi_{MRB}^W} \; 0 \; 0 \; \frac{\partial u_n^m}{\partial x_{MRB}^W} \; 0 \; 0 \; \frac{\partial u_n^m}{\partial y_{MRB}^W} \; 0 \; 0 \; \frac{\partial u_n^m}{\partial z_{MRB}^W} \; 0 \; 0 \right] . \tag{5.9.a}$$

Row $2 \times N \times (m-1) + 2 \times n$ of the linearized observation matrix is shown in Equation (5.9.b),

$$\left[\frac{\partial v_n^m}{\partial \psi_{MRB}^W} \; 0 \; 0 \; \frac{\partial v_n^m}{\partial \theta_{MRB}^W} \; 0 \; 0 \; \frac{\partial v_n^m}{\partial \phi_{MRB}^W} \; 0 \; 0 \; \frac{\partial v_n^m}{\partial x_{MRB}^W} \; 0 \; 0 \; \frac{\partial v_n^m}{\partial y_{MRB}^W} \; 0 \; 0 \; \frac{\partial v_n^m}{\partial z_{MRB}^W} \; 0 \; 0 \right] . \tag{5.9.b}$$

Since the observation model has no direct relationship with the state variables corresponding to the higher order terms, velocities and accelerations, the partial derivatives of the observations with respect to these terms are set to zero. A detailed derivation of the linearization of the observation model is found in Appendix E of this document.

5.2.3 Expected Errors and Covariance Models

5.2.3.1 System Dynamics Noise

The system dynamics model described in Equations (5.2) and (5.3), is a third order 6 DOF system. It was decided to account for inaccuracies in the dynamics model and perturbations to the system trajectory by assuming it could be modelled by white noise introduced into the dynamics model through the acceleration term. This is shown for a single degree of freedom in Equation (5.10),

$$\begin{bmatrix} x_k \\ \dot{x}_k \\ \ddot{x}_k \end{bmatrix} = \begin{bmatrix} 1 & \tau & \frac{\tau^2}{2} \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \\ \ddot{x}_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \omega_x \end{bmatrix} . \tag{5.10}$$

It follows from the dynamics model that this assumtion is the same for each degree of freedom with the expected noise levels being adjusted independently for each degree of freedom. The noise vector for a third order 6 DOF system is shown in Equation (5.11),

$$\omega_k = \begin{bmatrix} 0 & 0 & \omega_1 & 0 & 0 & \omega_2 & 0 & 0 & \omega_3 & 0 & 0 & \omega_4 & 0 & 0 & \omega_5 & 0 & 0 & \omega_6 \end{bmatrix}^T \quad . \tag{5.11}$$

The definition of the dynamics covariance from Chapter 4, is repeated here as Equation (5.12),

$$Q_k = E\langle \omega_k \omega_k{}^T \rangle \quad . \tag{5.12}$$

From this definition the dynamics covariance, Q_k, for a third order, 6 DOF system is an [18 x 18] square matrix. Just as there is no cross-coupling in the dynamics model and by the definition that the noise is uncorrelated, there are no cross-coupling terms in the dynamics covariance. The general form of the dynamics covariance matrix for a single degree of freedom is described in Equation (5.13.a),

$$\beta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_\omega^2 \end{bmatrix} \quad . \tag{5.13.a}$$

The single degree of freedom model is repeated for each additional degree of freedom as shown in Equation (5.13.b),

$$Q_k = \begin{bmatrix} \beta & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 & 0 & 0 \\ 0 & 0 & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta \end{bmatrix} \quad . \tag{5.13.b}$$

5.2.3.2 Observation Noise

Observation noise represents inaccuracies in the observation model and the noise in the signal coming from the sensor and data acquisition system. In the Argo Project, the assumption is made that two types of observable output exist: the horizontal image coordinate and the vertical image coordinate. A further assumption may be made that these two types of observable outputs are collected with different sensors, even thought they originate from the same image plane. This assumption is made to account for different horizontal and vertical resolutions in the camera. The

observation noise vector alternates with the same (u, v, u, v) pattern of the observation model as shown in Equation (5.14),

$$\nu_k = \begin{bmatrix} \xi^1_{1_H} & \xi^1_{1_V} & \xi^1_{2_H} & \xi^1_{2_V} & \cdots & \xi^m_{n_H} & \xi^m_{n_H} & \cdots & \xi^M_{N_H} & \xi^M_{N_V} \end{bmatrix}^T \quad . \tag{5.14}$$

The definition of the observation covariance from Chapter 4, is repeated here as Equation (5.15),

$$R_k = \mathrm{E}\langle \nu_k \nu_k{}^T \rangle \quad . \tag{5.15}$$

For the Argo Project, it is assumed that the variance in observations made in the horizontal and vertical axes are constant and equal,

$$\sigma^2_\xi = \sigma^2_{\xi_H} = \sigma^2_{\xi_V} \quad . \tag{5.16}$$

This assumption is made because the technical manual for the Qualisys camera system indicates one variance for both horizontal and vertical [Qualisys 92]. It is also felt that variance due to the observation model itself is likely to be similar for both axes. Each observed image coordinate has two components, horizontal and vertical. Therefore, the observation covariance matrix is square with the dimensions $[2 \cdot M \cdot N \times 2 \cdot M \cdot N]$, where M = number of cameras and N = number of markers on the target. The diagonal elements of this matrix are the variance of the observations, as shown in Equation (5.17),

$$R_k = \begin{bmatrix} \ldots & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_\xi & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_\xi & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_\xi & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_\xi & 0 \\ 0 & 0 & 0 & 0 & 0 & \ldots \end{bmatrix} \quad . \tag{5.17}$$

5.2.4 Data Acceptance and Rejection, and Pair Matching

The vector of estimated image coordinates, $\hat{g}_k{}^\dagger$, assumes that all markers are visible and that there are no ghost markers present. To be able to compute the error, $\hat{e}_k$, between the estimated,

$\hat{g}_k'$, and observed, $\tilde{g}_k$, image coordinates, the two vectors must have the same length (same number of observation) and have the same order. To ensure that the two vectors correspond, a data acceptance and rejection process coupled with pair matching and sorting routines are required. Figure 5.2 illustrates some of the difficulties associated with the acceptance and rejection process.

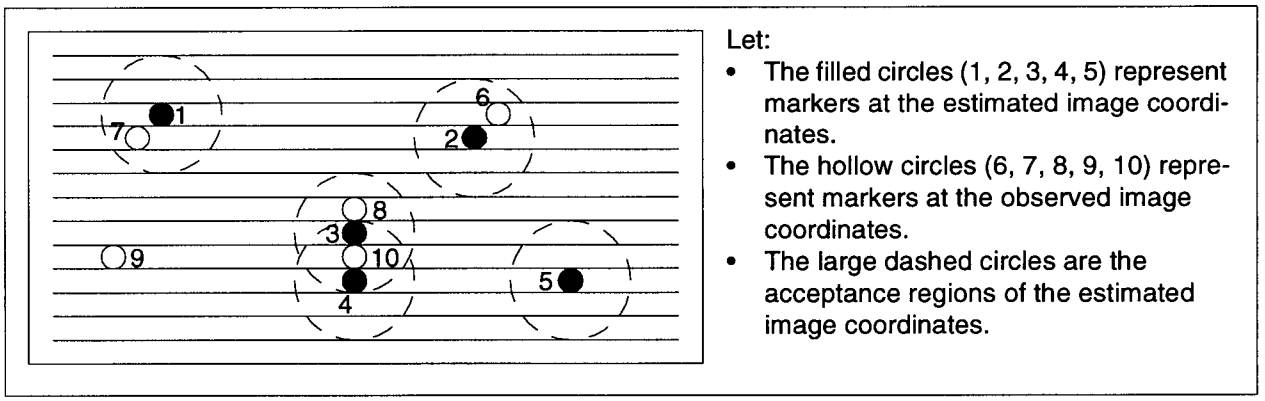


Figure 5.2: Image coordinates acceptance and rejection

Examining Figure 5.2, there are some obvious pairing assignments between the estimated image coordinates and the observed image coordinates. At the same time, it is also appearant that there are both estimated and observed image coordinates without pairs that must be rejected. Also, it illustrates an area of ambiguity, where the pair matching is not conclusive. These markers will be rejected.

Initially all estimated and measured observations are assumed to be rejected. As observations are accepted they are so tagged. At the end of the process all observations that have not been accepted are filtered out.

The acceptance process begins by computing the radial distance between an estimate image coordinate and all of the observed image coordinates. This list of radial distance is then tested to determine how many distances fall inside a preset tolerance. Considering estimated marker #1, only one observed marker #7 falls into the circle of acceptance; therefore, they are temporarily accepted as a valid pair. For estimated marker #2 and observed marker #6, this is also the case. For esti-

mated marker #3 two observed markers, #8 and #10, fall inside the acceptance region; however, both can not be valid. In this situation the estimated marker #3 and both the observed markers #8 and #10 are rejected because the pair matching is inconclusive. Estimated marker #4, only has observed marker #10 inside the acceptance circle, but since marker #10 has already been rejected in a previous validation test, estimated marker #4 is also rejected. If marker #4 had been tested prior to testing marker #3, marker #4 would have been paired with marker #10. But when marker #10 failed because of the double acceptance for marker #3, both marker #10 and marker #4 would then be rejected. Estimated marker #5 has no corresponding observed marker, possibly due to occlusion; therefore, it is rejected. Any remaining observed markers, for example marker #9, that were not paired with a unique estimated marker or were not rejected because of overlap, are considered as ghosts and are summarily rejected. This leaves estimated marker #1 paired with observed marker #7 and estimated marker #2 paired with observed marker #6. All others in the image are rejected and removed from the observation vector along with the corresponding rows in the linearized observation model, C_k, and the observation covariance model, R_k. The last task is to ensure that the order of the incoming observation vector, $\tilde{g}_k$, matches that of the estimated observation vector, $\hat{g}_k^{\dagger}$. Image coordinates in a video image are referenced left to right, top to bottom, with the origin of the image plane in the upper left corner. Referring back to Figure 5.2, the order for the estimated markers is #1 follows #2; however, their observed pairs are in the reverse order, #6 precedes #7 because #6 is one row above #7. As a result the incoming observation vector, $\tilde{g}_k$, is sorted to match the order of the estimated observation vector, $\hat{g}_k^{\dagger}$, so that the error between measured observations and estimated observations can be computed. This process is repeated for each separate field of view in a given frame. For example, for a three camera system there are three images per frame. A more detailed discussion of the data acceptance and rejection, pair matching, and sorting may be found in Appendix F.

5.2.5 Initialization of Tracker and State Error Covariance

To initialize the tracking process, it is necessary to provide an initial state estimate, $\hat{s}_0$, and state error covariance, P_0. Every effort should be made to provide the best possible initial estimates as this will help the EKF establish track sooner. The initial dynamics model, A_0, the dynamics covariance, Q_0, and the observation covariance, R_1, are considered constant and are defined based on known and estimated parameters. All other terms in the tracking algorithm are computed from these basic terms.

During experimental and simulation tests conducted with the Argo Project, the target is held still for the first ten seconds to establish the initial pose. The initial pose is computed by taking the observation model, Equations (5.6.a) through (5.6.e), and solving for the six pose terms, $(\psi^W_{MRB}, \theta^W_{MRB}, \phi^W_{MRB}, x^W_{MRB}, y^W_{MRB}, z^W_{MRB})$, using a least squares method. When determining the initial pose, the target is assumed to be stationary; therefore, the initial velocity and acceleration terms of pose are zero. As a result, the initial state estimate, $\hat{s}_0$, is only a function of the initial pose of the MRB.

The initial state error covariance, P_0, is based on the dynamics model and the expected dynamics and observation noise. Since the state error covariance is a self adjusting entity within the tracker, any inaccuracies in this initial guess will be self corrected with the execution of the tracker. For the models used in this project, it was determined that an appropriate initial estimate of the state error covariance, P_0, is as shown in Equations (5.18.a) and (5.18.b). A detailed derivation of this initial estimate of the state error covariance is given in Appendix D. The initial state error covariance for a single degree of freedom is:

$$\Gamma = \begin{bmatrix} \sigma_\xi^2 & \sigma_\xi^2 & \frac{\sigma_\xi^2}{\tau^2} \\ \sigma_\xi^2 & 2\frac{\sigma_\xi^2}{\tau^2} & 3\frac{\sigma_\xi^2}{\tau^3} \\ \frac{\sigma_\xi^2}{\tau^2} & 3\frac{\sigma_\xi^2}{\tau^3} & \sigma_\omega^2 + 6\frac{\sigma_\xi^2}{\tau^4} \end{bmatrix} . \tag{5.18.a}$$

The single degree of freedom is repeated for each additional degree of freedom. The 6 DOF initial state error covariance model is:

$$P_0 = \begin{bmatrix} \Gamma & 0 & 0 & 0 & 0 & 0 \\ 0 & \Gamma & 0 & 0 & 0 & 0 \\ 0 & 0 & \Gamma & 0 & 0 & 0 \\ 0 & 0 & 0 & \Gamma & 0 & 0 \\ 0 & 0 & 0 & 0 & \Gamma & 0 \\ 0 & 0 & 0 & 0 & 0 & \Gamma \end{bmatrix} . \tag{5.18.b}$$

5.2.6 Incoming Data and Simulated Data Generation

Figure 5.1 shows that the Argo Project may be configured for two possible sources for the data stream of image coordinates. The intended use of the program is to process recorded experimental data gathered from a set of synchronized video cameras.

However, for testing purposes, simulated data is fed to the tracker in place of recorded experimental data. The simulated data stream is similar in form to a data stream captured from the camera system and is generated independently of the main tracking algorithm. It uses only the target geometry and the current frame number as inputs to this subroutine. The process for generating simulated data is first to generate the pose of the MRB in the world coordinate system for the current image frame. Using this pose, the positions of markers in the MRB coordinate system are transformed into the world coordinate system. With the marker locations in the world coordinate system they are then projected onto the image plane. The current version of the simulator uses a

simple plane perspective projection camera model to generate the image coordinates. It is planned to upgrade this model to be more representative of the data gathered with real cameras. This camera model is completely separate from the one used in the observation model, although it is very similar in operation.

5.3 Hardware Implementation

The hardware components necessary for the Argo Project are:

- a host computer.
- video processors that act as frame grabbers and also do preliminary image processing.
- synchronized video cameras equipped with strobes.
- a target of known geometry attached to the MRB.

A schematic of the configuration of these components is shown in Figure 5.3.

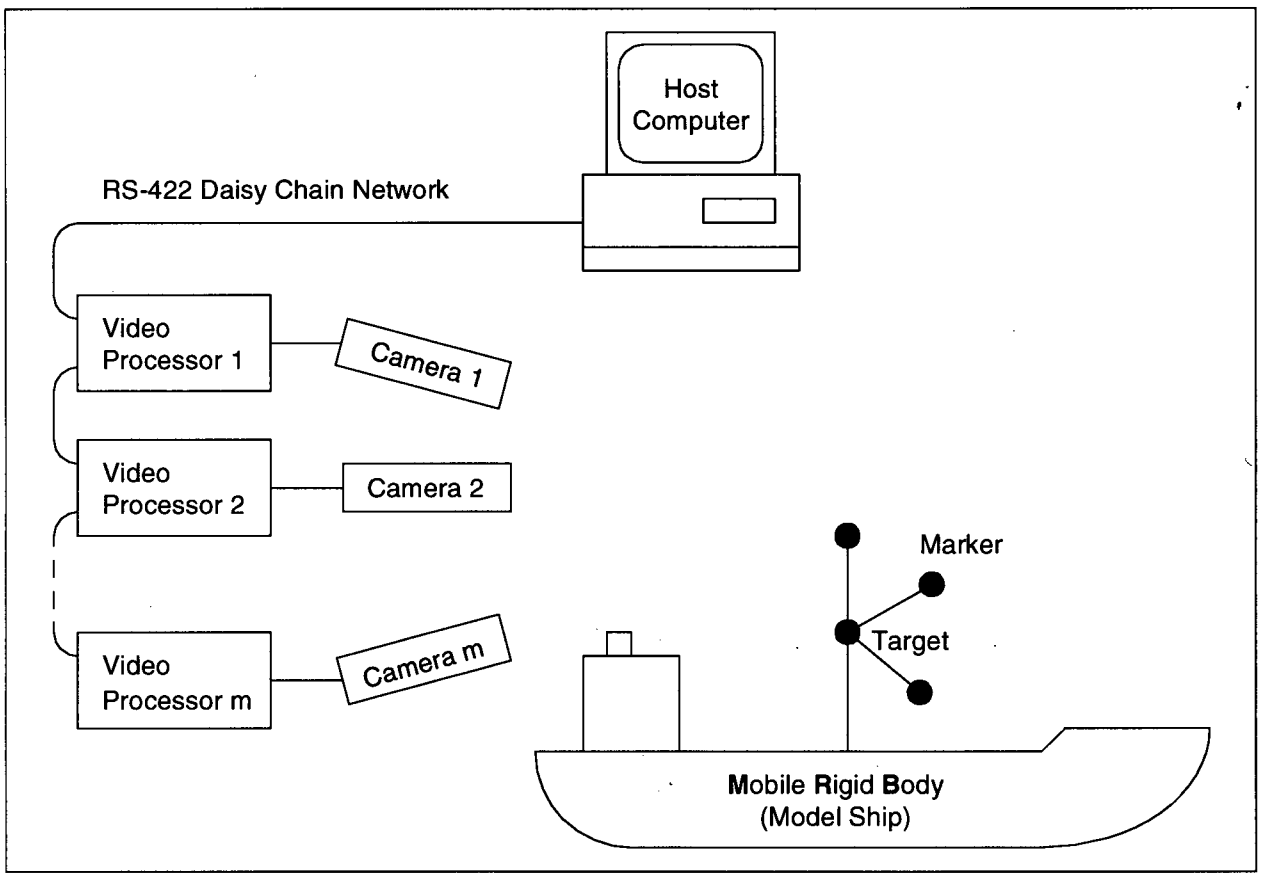


Figure 5.3: Argo Project Hardware Configuration

5.3.1 Video Cameras and Processors

The video cameras and video processor hardware used for this research is the Qualisys MacReflex™ system. The reasons for this selection are presented in Chapter 2. The Qualisys MacReflex™ system supports one to seven video processor and camera pairs linked to the host computer through a single local area network.

The video camera used in the Qualisys Mac Reflex™ system is a modified industrial CCD video camera with PAL video encoding. The system has sample rates that can be set up to a maximum of 50 Hz, twice the frame rate of the PAL standard of 25 frames/sec. Qualisys employs a trick that effectively doubles the frame rate of the camera by treating each field of a frame as a separate image. A full interlaced video frame is comprised of two fields. A field is every other line of a full interlaced video frame, in effect the first field would be all the odd numbered lines in a frame and the second field is all the even number lines of the same frame. When these two fields are meshed together the full frame is formed [Greaves]. This results in a 604(H) X 294(V) image, which has the same horizontal resolution as a full frame but only half the vertical resolution of a full frame [Qualisys 92]. The camera passes to the video processor a nearly all black image except for some high intensity bright spots that most likely represent markers visible in the scene. To achieve this the camera has four features not normally found on industrial video cameras:

- A flash, made of directional IR LEDs clustered around the lens, is used to flood the field of view (FOV) for that camera with IR light. The retroreflective nature of the markers reflect this light back in the same direction it originated. Since the flash surrounds the lens, most of the returning light is captured by the lens lowering the power required for the flash. The net effect of this is that high intensity spots in the image likely correspond to markers in the scene.
- The lens is fitted with an IR band pass filter. Therefore, only IR light sources will be seen in the image. These light sources can be reflections from the IR flash or from IR sources such as LEDs or LASERs.
- A high speed electronic shutter, $1/1250^{th}$ of a second [Qualisys 92] synchronized with the flash is used. This short exposure time allows only high intensity light sources to register in the

image.

- Variable gain is used to control light amplification. High intensity light sources in the image are reduced to normal levels and all others elements of the picture become dark. [Qualisys 92]

Together, these features eliminate everything from the image that is not a strong IR source or reflector. This eliminates most concerns about ambient lighting and allows researchers to work safely in a well lit environment.

High intensity circular patches in the image are perceived to be images of markers in the scene. However, this is not guaranteed as wave crests and shiny surfaces such as a plated bolt or a piece of aluminium in the scene may also return a signal, termed as a ghost marker. The determination of whether the return is a valid marker is handled by the data acceptance-rejection tests, mentioned previously in Section 5.2.4.

The proprietary Qualisys video processor acts as camera controller and power supply, frame grabber, and performs feature extraction using built in firmware. The video processor first grabs the high contrast black and white image and identifies light blobs in the image as potential features. Then, with subpixel interpolation, it determines the centroid, height and width of each potential feature. These data are captured to the computer with the acquisition and camera control software where they are stored or processed.

An Apple MacIntosh computer is the intended host computer for the Qualisys Mac Reflex™ system. As a result, communication between the video processors and with the host computer is done via a daisy chained RS-422 serial connection, native to the Apple MacIntosh. This daisy chained network runs at the maximum 500 kbaud set in the video processor hardware. The relatively low band width of this serial RS-422 network limits the number of markers that can be tracked at any one time.

The host computer uses the serial network to send a reset command to all the video processors and an acquire command to the master video processor. The master video processor receives the

acquire command and starts an internal clock, with an adjustable frequency (default is 50 Hz), that in turn triggers the slave units to simultaneously take a picture of the scene. Each video processor simultaneously captures an image, and through dedicated hardware and internal software, returns the image coordinates of the centroid of each image blob above a preset intensity threshold along with the corresponding height and width. After completing its image processing, it will transmit to the host computer the camera identification code, the number of bright spots found in the image, the frame number and the data for each bright spot in the image. This data is used for further data reduction and tracking by software on the host computer.

Additional details on the communication protocols, the data stream structure and the physical characteristics of the Qualisys Mac Reflex™ system hardware can be found in their technical manuals [Qualisys 92].

Recently, Qualisys has made available 60 Hz systems that are based on the NTSC standard and they now support the PC environment. They are also considering replacing the RS-422 serial connection with a higher speed network to increase the data bandwidth, allowing for more markers to be tracked.

5.3.2 Camera Placement

The placement of cameras to observe the motion of an MRB is important for a successful test. There are two components to camera placement, the position and the orientation of the camera. There are several points to consider when choosing camera placement.

The test volume that the MRB travels in is defined by the field of view (FOV) of the multiple video cameras monitoring the scene. Selecting the FOV, through camera placement and the lens focal length, is a compromise between the desire to have high resolution and to have a large test volume. The larger the field of view of the camera, by placing it farther away from the scene or using a wide angle lens, increases the test volume and lowers the resolution with which the scene is observed. Conversely, a narrow, close field of view forces the testing volume to be small but

increases the resolution with which the scene is observed.

To avoid distortion of the image the camera should be placed so that during a test the target appears in the centre of the image. Because a component of lens distortion is a function of radius, the farther away the target is from the focal axis, the more pronounced the distortion becomes.

The proper camera orientation can help eliminate or alleviate some of these problems. Human beings like to see images right way up and assume automatically that an image is unless there are other indicators to the contrary. However, the computer has no preconceived notions of how an image should appear. Because of this a camera may be placed on its side or at an odd angle if it helps to view the scene better. For example, consider a regular video image, that is wider than it is tall, used to observe a target that has most of its travel in the vertical axis. It would be more suitable to place the camera on its side in order to align the image axis with the greatest resolution to the axis of greatest interest or of largest displacement in the image.

Also, efforts should be made to minimize or eliminate obstructing from the view of a camera any part of the scene that the MRB is likely to travel.

5.3.3 Host Computer

The host computer collects the data from the video processor units and runs the tracking software. The host computer selected for the Argo Project is an Intel x86 processor family based personal computer (PC). The reasons for this selection were to use existing computer equipment and to maintain compatibility with the system under development at NRC-IMD. A Quatech MPA 200 RS-422 card was used to interface the Qualisys video processors to the host computer.

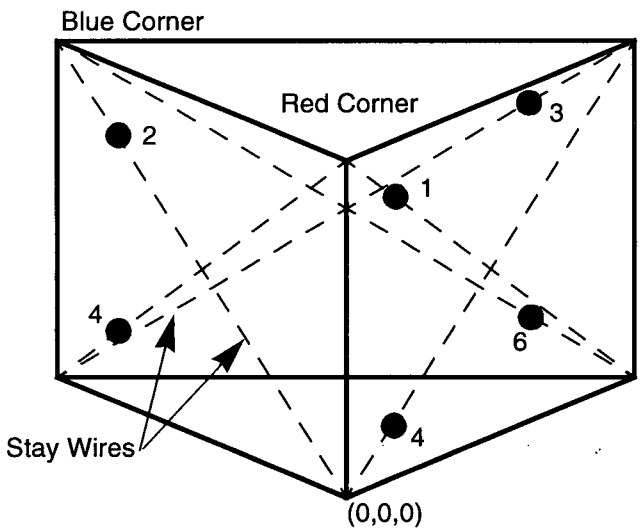
5.3.4 Target

The target is comprised of a minimum of four non-coplanar, spherical, retroreflective markers at known locations in the 3D space of the target coordinate system. The markers used in the target are either spherical, hemispherical or disks that are roughly the size of ping pong balls and are coated in a retroreflective paint which is highly reflective of light in the near-IR wavelengths. The

sphere is the preferred marker shape as the position of its observed centroid is fixed for all angles of view. In the case of a hemisphere, viewing it from the side will shift the centroid away from the flat edge by a factor of 0.4244 × the radius of the hemisphere [Gieck 90]. Disk markers, on the other hand, can not be viewed from the side at all.

The markers are mounted to a frame to form a target. The target frame structural members should be painted flat black with no shiny fastners or brackets left exposed to try and eliminate the possibility of stray reflections that could result in ghost markers. Similar care should be taken when preparing the MRB for testing. Although, there is no need to paint the MRB flat black.

The target is attached to the MRB with a known pose relative to the MRB's frame of reference. This allows for the mapping of the marker locations into the MRB reference frame. The target geometry, the marker locations in the target coordinate system, should be carefully selected to avoid symmetry in the layout. This is necessary to avoid confusion when attempting to identify markers in a image. A poor example of a target is the Qualisys™ calibration frame that has six markers defining the corners of a vertical equilateral triangular prism, shown in Figure 5.4. In an image, this geometry appears the same upside down or when rotated ±120° about its vertical axis. In ad-



Qualisys Calibration Target

Marker locations (units are mm)

	X	Y	Z
1	(123,	0,	662)
2	(446,	773,	662)
3	(954,	107,	662)
4	(123,	0,	88)
5	(446,	773,	88)
6	(954,	107,	88)

[Qualisys ??]

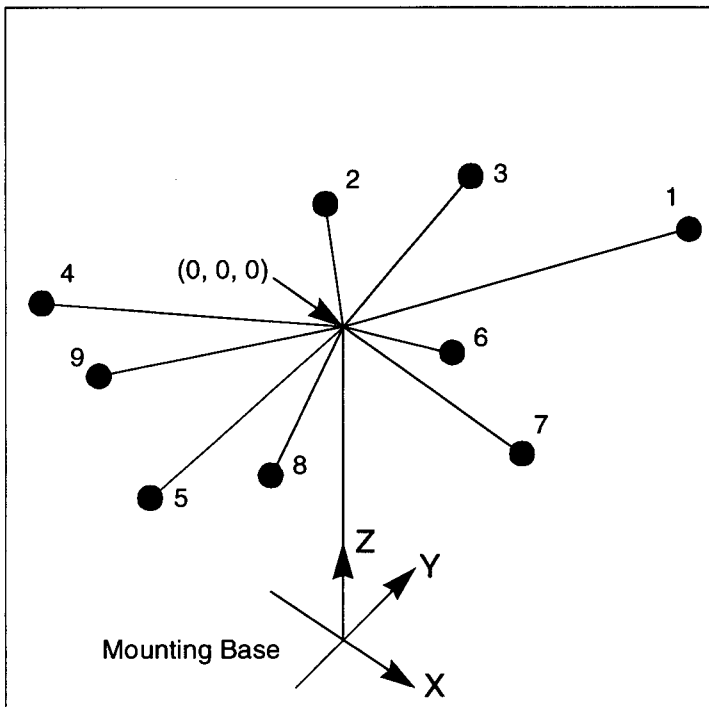
Figure 5.4: Qualisys Calibration Target Geometry

dition the markers on this target frame are equally spaced making it even more difficult. This is the target frame that was used in the experimental testing of this project. A more suitable target is one where the markers are randomly placed and the structure of the target does not occlude the markers. An example of this is the NRC tree target, shown in Figure 5.5.

5.4 Software Implementation

The motion tracking software component of this system is a set of subroutines collectively known as the program Argo. The program tracks the 6 DOF rigid body motion of an arbitrary MRB with a known target attached to it. This program has two main modes of operation: simulation and processing experimental data. The tracking algorithm implemented is a scalable modified EKF loop with selectable data sources: computer generated image coordinates or image coordinates captured from the Qualisys hardware.

The Argo program is not currently stand alone and operates in the MATLAB™ environment. MATLAB™ is an interpreter-based high level programming environment that is relatively user



NRC-IMD Target Geometry

Marker locations (units are mm)

	X	Y	Z
1	(533,	4,	510)
2	(-19,	14,	432)
3	(1,	421,	404)
4	(-584,	-2,	2)
5	(-3,	-736,	-3)
6	(2,	207,	-190)
7	(215,	0,	-194)
8	(2,	-322,	-302)
9	(-420	0,	-407)

[Sullivan PC]

Figure 5.5: NRC Tree Target Geometry

friendly and permits efficient coding of algorithms. The disadvantages of this environment are that it can not produce a stand alone product and that the program's execution may be slower then an optimized and compiled version of the same program. To turn this research effort into a useable product, the software will have to be ported to a compiler based programming language such as C, C++ or Pascal. There are four reasons for this: (*i*) the program must be stand alone, (*ii*) the program must be able to interface with the system hardware to read the data streams from the video processor units, (*iii*) the program must be able to operate under a real-time operating system, and (*iv*) security against theft of intellectual property and unauthorized modification of the program is greatly improved with a compiled version because the source code is not directly available to the user.

The program's different operation modes are controlled by a configuration vector that contains option flags. These option flags control everything from where the data stream originates, which graphs to generate, which diagnostic routines to run, and other internal house keeping chores. The program queries the user with prompts such as what data file to load and where to store the processed data. This program can also operate in a batch mode where the program queries a batch file rather than the user, freeing them from having to be present while the computer processes data.

5.4.1 Interfacing with Qualisys hardware

Since, Argo has been written in the MATLAB™ environment, it is not possible to communicate with the Qualisys hardware in real time; therefore, existing external programs were used to capture and store the image coordinates generated by the video processors. The program used to capture the image coordinates and then store them to binary data files was written and developed at NRC-IMD [Sullivan PC]. This program is part of their in house research effort in motion tracking of model ships. An additional software utility furnished by NRC-IMD translates these binary data files to ASCII text files which can then read by the Argo program.

5.4.2 Program Initialization

The Argo program starts with a multi-step internal initialization process that:

- clears system memory;
- loads the configuration vector containing option flag values that control the operation of the program;
- if in batch mode, it loads the batch file and deactivates all further user prompts (this option allows the program to run for extended periods of time without direct supervision by the user);
- starts the screen activity log to record all user inputs and program text outputs
- logs the start time;
- defines global and system variables specific to the mode of operation;
- calibrates the cameras for the session;
- if processing recorded data, it loads the image coordinate data stream history into memory.

5.4.3 The Main Program Loop

To be able to process a series of image sequences,the tracking loop is nested inside a main loop that indexes through a set of simulations or data files to processed. If the source of the image sequence is a recorded data file, then it is loaded into memory. If triangulation of a stationary target is being done, the marker correspondence lists for each camera are loaded. These correspondence lists have to be determined by hand in advance of running this option. The target geometry, target pose with respect to MRB space and marker locations in target space, are loaded and the marker locations are transformed into MRB space. With all the appropriate data loaded, there are three options: generate a scatter plot of the entire image sequence, triangulate a stationary object, or track a moving object.

At the beginning of tracking any sequence, the initial conditions for the EKF are set based on a triangulation of the starting pose of the target. The EKF tracking loop is indexed with the current frame number. At the beginning of each cycle through the loop, a request is made for the next frame, and all the image coordinates associated with that frame. If the program is running in sim-

ulation mode the frame number is sent to the Softcam program and a set of image coordinates is returned; if running in data processing mode, the next frame is pulled from the frame history. Once the data stream has been preprocessed by the data acceptance and rejection routines, it is then passed onto the EKF loop for current state estimation. Each state vector is recorded and stored in a file for later analysis.

5.4.4 Simulation and subroutine Softcam

Softcam is a subprogram inside the Argo program that generates a data stream of image coordinates with the same form as those captured from the hardware. Softcam, although contained in the main program, is independent in operation. The interaction between Softcam and Argo is limited to Argo passing the current frame number as an input to Softcam and receiving the image coordinate vector as the output. Using the frame number and a known frame rate, a subroutine internal to Softcam uses a set of hard coded equations of motion for each degree of freedom to generate the next pose that the MRB will experience in the simulation. This generated pose is then used to compute the 3D position of the markers in world space. The camera model used to generate the image coordinates of these markers is plane perspective projection, a simple pin hole camera model with no lens distortion. This camera model is completely independent of the one that is part of the observation model in the EKF tracker.

5.4.5 Diagnostic subroutines

The Argo program contains several diagnostic service routines to help in the assessment of new algorithms. These routines would have to be removed from a final product because they greatly impede computational efficiency.

A scatter plot of the full image sequence allows the operator to check for ghost images in the image sequence. This plot is also useful for generating the image correspondence lists needed for triangulation and calibration.

Triangulation is done as a check of calibration and target placement. Triangulation can be done

on a single or averaged image if the correspondence lists are provided. This does not necessarily preclude triangulating the locations of markers on a moving target but a correspondence list would have to be generated for each image. As a result, this is usually reserved for stationary targets. As a double check, the inter-marker distances for the known geometry and the triangulated markers are compared and non-dimensionalized to give a single measure of error in the triangulation.

5.5 Summary

The Argo Project, using an EKF based tracking algorithm, tracks the six degrees of freedom motion of an MRB following a continuous and random trajectory with data gathered from a set of synchronized video cameras. The tracking software takes the stream of image coordinates corresponding to the centroids of potential markers as input to the tracking algorithm. If a potential marker in the image can not be conclusively identified through a series of tests and paired with its corresponding estimate, it is rejected by the data acceptance and rejection routine.

This software is flexible in its design because it can track the trajectory of an MRB from a sequence of image coordinates, independent of the source. Any image capture system, film or video, that will return the centroids of potential markers in the image is sufficient. Certain dependencies will have to be built into the software for specific image capture systems; however, these modifications are external to the core of the program that will remain unchanged.

The software component of this system has been developed in the MATLAB™ environment for off line execution. There are plans to port this program to the C language so that it can be compiled for real-time operation.

The advantages to this system are:

- Any number of cameras, from one to as many as the hardware can support, can be used to study a scene. The more cameras that are used, the greater the reduction in the error in the state estimate.

- Ghost images that may appear in the image are handled.
- Occluded markers are handled provided that not too many are hidden from view. Since a single camera can be supported, if a marker is only visible in one view, the image coordinates are still valid data points and they are included into the measurement vector.
- Combining the above listed advantages also allows for the possibility of an increased test volume. With a multiple camera system the target can go out of the view of one camera and come into the view of another while constantly remaining in the view of a third camera and still be tracked. This means that test resolution need not be sacrificed because of a large test volume. Many cameras could be combined together to cover the large area and not compromise on the resolution of the data.

Chapter 6: Experimental and Simulation Results

6.1 Introduction

The Argo Project software was tested by both simulation and processing of data collected from physical experiments. Simulations have been ongoing through out the project to evaluate how new features and refinements affect the system's performance. During the summer of 1996, physical experimental tests were conducted at the NRC-IMD facility in St. John's, Newfoundland, in conjunction with C-CORE's Intelligent System Division. The data collected in these tests were subsequently post-processed using the Argo software to track the motions of the moving target.

The results of experimental and simulation testing of the Argo Project indicate that the concepts behind this research effort do work, but need further refinement to make this work into a practical tool for towing tanks. It appears that the tracked oscillations tends to lag physical oscillations in excess of 1/2 hertz, while frequencies higher than 20 hertz are filtered out of the signal.

6.2 Evaluation by Simulation

Simulation tests with the Argo software were done, by generating a stream of image coordinates in a form similar to the data stream, that comes from the video processor units. These simulated image coordinates are generated by:

(i) calculating the pose of the simulated MRB based on a set of known equations of motion where the frame number is used as the time variable;

(ii) mapping the locations of the target markers from the MRB coordinate system to the world coordinate system;

(iii) projecting the marker positions in the world coordinate system onto the image plane, through a camera model separate from the one used in the tracking algorithm.

This procedure functions entirely separate from the main tracking program. The only common

information between the two programs are the current frame number, the target geometry, and the pose of the target with respect to the MRB's coordinate system.

The physical setup represented by the simulation discussed herein is depicted in Figure 6.1[1]. The test volume is defined by three cameras observing the scene. The cameras are modelled as 1/2 inch CCD video cameras, each with an image plane measuring 6.4 mm wide by 4.8 mm tall. The camera located at (0 m, 0 m, 4 m) is fitted with a 10.0 mm lens with a horizontal field of view of 35.5°. The remaining two cameras are fitted with 8.5 mm lenses that have a horizontal field of view 31.5°. All cameras have a downward looking angle of roughly 20° below the horizon. The target used in this test is the NRC Tree Target, shown in Figure 5.5. This target is attached to the MRB with a position of (0 m, 0 m, 1 m) and an orientation of (0°, 0°, 0°), relative to the MRB's coordinate system. With a combination of three cameras ($M = 3$) and nine markers on the target ($N = 9$), the system has a maximum of ($2 \cdot M \cdot N$) 54 constraint equations defining the pose of the MRB in the world coordinate system.

The results of this simulation are presented in Figures 6.2 through 6.7. To avoid cluttering the figures, the units of the data are listed in Table 3.1. In these figures, the solid lines represent the known inputs and the dashed lines denote the estimated tracked output of the EKF. The particulars of this simulation are:

- Simulation duration is 60 seconds;
- Sampling rate is 100 frames/second; therefore, the simulation is represented by 6000 frames;
- The generated roll motion is corrupted with white noise, with a signal to noise ratio of 75;
- Roll motion (rotation about the x axis) is generated by using a sinusoidal motion, with an amplitude of 5 degrees, at a frequency of 0.10 hertz, about 0 degrees mean angle;
- Pitch motion (rotation about the y axis) is generated by using a sinusoidal motion, with an amplitude of 10 degrees, at a frequency of 0.13 hertz, about 0 degrees mean angle;
- Yaw (rotation about the z axis) is held fixed at 10 degrees;
- Surge motion (motion along the x axis of the world coordinate system) is generated using a

1. Note: figures in this chapter are included at the end of the chapter due to their size and number

sinusoidal motion with an amplitude of 2.5 metres, at a frequency 0.05 hertz, and with an offset of 8.5 metres;

- Sway motion (motion along the y axis of the world coordinate system) is generated using a sinusoidal motion with an amplitude of 2.0 metres, at a frequency of 0.05 Hz, and with an off-set of 9 metres;
- Heave motion (motion along the z axis of the world coordinate system) is generated using a sinusoidal motion with an amplitude of 0.4 metres, at a frequency 0.25 hertz, and with an offset of 0.4 metres.

The results show that the tracked pose trajectory closely matches that of the known trajectory, as shown in Figure 6.2. There is some error in each channel of pose, but it should be noted that the error is at least one order of magnitude less than the signal being observed, as shown in Figure 6.3. The roll error shows the high frequency noise contained in the input signal, indicating that the EKF filtered it out of the signal. The yaw error shows how error in one channel can effect another. Prior to frame 500, the target is essentially motionless, with the exception of a slow drift in the roll and pitch terms. During this time period the error in yaw is minimal, after frame 500 the target starts to move, and immediately the level of yaw error increase. This increase in error is an artifact of the EKF tracking process. The low frequency motions in the x-y plane (surge and sway) show excellent position tracking. The higher frequency motion of heave shows a slight over-shoot compared with the known heave motion. It is felt that this error could be reduced by adjusting the gain of the EKF.

This simulation also tested the robustness of the Detection and Rejection function of the Argo program as marker four ($n = 4$) moved in and out of view of camera one ($m = 1$). The system successfully managed to deal with a marker that disappeared from view and then reappeared later on. The Detection and Rejection function also rejected data if it could not be conclusively identified for purposes of correspondence mapping.

6.3 Experimental Testing

The physical experimental testing of the Argo Project consisted of using combinations of two to four Qualisys MacReflex™ Motion Sensor systems observing a target attached to the bottom of a swinging pendulum, as shown in Figure 6.8.

The data streams from the Qualisys video processors, are captured to the memory in the host computer and then stored in binary coded files. These files were later translated to ASCII format so that they could be read into the MATLAB™ programming environment for processing by the Argo software. The software used for controlling the Qualisys hardware and capturing the incoming data streams, as well as the translations software was provided by NRC-IMD.

The Qualisys calibration frame was used as the target, and is shown in Figure 5.4. Provided that all six markers on the target ($N = 6$) are visible, in all four video cameras ($M = 4$), then the maximum number of constraint equations defining the pose of the target is $(2 \cdot M \cdot N)$ 48.

The pendulum used in these experiments was an existing aluminium swing-frame, with the Qualisys calibration frame attached to the base of the frame, as described in Figures 6.9. This swing-frame, originally designed to determine the radius of gyration of ship models, was made available by NRC-IMD. Lead bars were added to the base of the swing-frame, to increase the swing period and to decrease the damping effect. By adding the lead bars the effective length of the pendulum was approximately 2.23 metres. The pivot axis of the pendulum was 4.0 metres above the floor.

An accelerometer based inclinometer were used to log the angle of the swing-frame. The inclinometer data were sampled at 100 Hz using a data acquisition system that was completely separate from, and not synchronized with, that of the camera system. The data gathered from the inclinometer can be used to corroborate the amplitude, frequency, and rate of decay of the tracked results. Unfortunately because the video stream and accelerometer data captures were not synchro-

nized it is not possible to evaluate the phase of the tracked output compared with a secondary independent source output.

6.3.1 Experimental Procedure

At the beginning of each test session, the cameras recorded images of the target with the pendulum held stationary, in the plumb condition, for calibration purposes. Also when ever any of the cameras were moved either intentionally or accidentally, or if the cameras were left unsupervised, images of the stationary target were recorded for recalibration of the cameras. With the calibration images logged, the swing-frame was then setup for either free and forced swing tests.

For the static tests, the swing-frame was held fixed at a known angle. This test was done primarily for calibration purposes but was also used for triangulation and pose extraction tests. For the majority of the static tests all four cameras were used; however, some of the tests were done with two and three camera configurations.

For free swing tests the swing-frame was set to an arbitrary start angle, which was recorded with a portable digital inclinometer. The swing-frame was held in place by a string tied to a fixed anchor point. The camera and the accelerometer data acquisition systems were started and logged data for 5 to 15 seconds prior to the release of the swing-frame. The string was then cut to release the pendulum. This method was simple, did not interfere with the pendulum, and did not introduce any artificial motions.

For the forced swing tests, the pendulum was pushed by hand in an irregular fashion producing a psuedo-random trajectory.

In both forced and free dynamic tests, the camera and the accelerometer data acquisition systems were started and logged data for 5 to 15 seconds while the pendulum was stationary to establish a starting state for the EKF.

To test different modes of operation and the error handling capabilities of the software, different test configurations were tried. Some of the tests were conducted using combinations of two

cameras and three cameras, even thought all four cameras were available. This was done, to try and observe if the additional data from more than two cameras, improved the ability to track the target. The tests were inconclusive and should be repeated both in simulation and by additional physical tests. Some of the reasons for not being able to draw any conclusions from these tests are that, not enough runs could be analysed, because the software was not able to track the trajectories of many tests runs. Occlusions were also tested by intentionally placing panels between a camera and the target. As the target swung some of the markers would move in and out view of the camera. Error handling for ghost markers was also tested by intentionally placing an extra marker in the scene for some of the tests. To test a random trajectory, a few tests were conducted where the swing-frame motion was forced by pushing the swing-frame with an erratic motion.

6.3.2 Experimental Results

During the three days of experimental testing, there were a total of 78 tests recorded. The 78 tests can be broken down into the follow three classes: 50 were static tests, 23 were dynamic free swing tests (regular motion), and 5 were dynamic forced swing tests (irregular motion).

Unfortunately, of the 28 dynamic tests recorded, the Argo software successfully tracked only two dynamic free swing tests. It is believed that the failure to track the pendlum's motion, for the majority of the tests, can be attibuted to three reasons:

(i) The camera model used in the observation model, do not include a lens distortion model. This is significant because, as the target moves away from the focal axis of the camera, the radial lens distortion comes into effect.

(ii) The camera model in the observation model was likely not sufficiently calibrated for the testing being done. This oversight in calibration was not realized until after the testing was completed and further testing was not possible. At the time of the testing the level of required calibration was still unknown and as a result, only a basic calibration was done. The calibration used was based on image sequences of the target, attached to the swing-frame hanging at a zero degree incline. This method yielded a calibrated test volume that was too small for the tests being conducted. As the target moved away from the region of the image that was cali-

brated, the estimated observations became less accurate. In the future the entire test volume should be calibrated.

(iii) The EKF may not have been tuned adequately for the system under observation.

As a result, only runs that did not see the target move far away from the calibrated region of the image, near the centre of the image, were tracked with the current version of the software. Before any further testing is attempted all three of the above deficiencies must be dealt with.

As an example, of the output from the Argo tracking software, for one of the successful natural free swing tests is shown in Figure 6.10, 6.11 and 6.12. The test results presented in Figures 6.10 through 6.12 are for a swing-frame swung in the y-z plane, observed by four cameras with no occlusions and no additional ghost markers. Again, to avoid cluttering the figures the units of the data are listed in Table 3.1.

Figure 6.10, shows the time history of the pose of the target. With the swing-frame constrained to swing in the y-z plane, the only expected motions are a rotation about the x axis (roll) and linear displacements along the y and z axes. The remaining pose terms are considered to be fixed at their initial values. The tracked results are as expected for a free swinging pendulum. The tracked roll (rotation about the x axis) and sway (movement along the y axis) motions are sinusoidal with an exponential decay, as expected.

The additional data gathered from the inclinometer, an independent secondary source, can partially corroborate the tracked output. Figure 6.13, shows that amplitude, decay rate, and frequency of the tracked roll angle of the target compared with the measured roll angle are very closely matched. The decay function superimposed on both plots in Figure 6.13, is represented in Equation (6.1):

$$\psi_D = \pm 8.13 e^{-0.0029\tau} \quad , \tag{6.1}$$

The decay rate of -0.0029 was determined by inspection only and not computed using energy meth-

ods. Figure 6.14 shows that the swing frequency of tracked roll data and the measured data align at approximately 0.34 Hz.

The tracked heave (movement along the z axis) motions are more significant as they indicate that the motion being tracked is that of a pendulum. The troughs of the heave signal are flattened at the same level for each cycle, while the peaks have a naturally damped exponential decay. This is the expected motion for heave, because for a pendulum of fixed length the lowest point of the swing is constant and does not change as it slows. The effect of forcing the troughs of the heave signal to have the same value has the results that the heave signal contains a small secondary higher frequency component. Figure 6.15, a plot of the heave power spectrum vs frequency, shows the heave signal and its first harmonic. The main frequency of approximately 0.34 Hz corresponds to the swing of the pendulum, and the frequency of the first harmonic at approximately 0.68 Hz, with a smaller energy, resulting from the truncated troughs of the heave signal.

The small ripples in the yaw motion of the target, can be attributed to small vibrations that may be caused from misalignment in the two pivot points of the pendulum. This motion falls within experimental error and is considered as noise.

Examining the rates of pose, shown in Figure 6.11, the roll and sway terms are obviously the derivatives of the corresponding terms in Figure 6.10. The orientation and position terms of roll and sway respectively, can be described by a negative cosine function with exponential decay. As expected, the derivative of these terms can be represented by a positive sine function with an exponential decay. The rate of pitch and yaw have increased noise levels, but this is expected when differentiating a signal. The heave velocity signal is of the greatest interest because the small step at zero speed corresponds to the flattened trough in the heave displacement signal.

Figure 6.12 shows the accelerations of the target. The heave acceleration signal is the most significant to note because it shows the changes in acceleration corresponding to the flattened trough of the displacement signal.

6.4 Summary

While the Argo software has its limitations it has been demonstrated to function both in simulation and processing experimental data gathered from physical testing. The tracking algorithm can track low frequency (<1/2 Hz), large amplitude (<10m or <30°) motion for all six degrees of freedom. The dynamic response of the tracker to changes in trajectory is slow, causing it to lag the system that is under observation. Tuning the EKF tracker by refining the expected noise levels in the dynamics and observation models, could increase the dynamic response of the Argo Project from 1/2 Hertz to 5 Hertz. Another possible modification that could improve the dynamic response of the tracker is to replace the linear dynamics model in the EKF with one that is based on the equations of motion for a pendulum. This effort has the draw back that the tracker will loose the ability to follow a non-oscillatory trajectory. This requires a compromise between an all purpose tracker that will have a lower dynamic response and one with a high dynamic response that is designed for a specific task.

Table 3.1: Units of Figures 6.1 - Figures 6.13

variable description	variables	units
angular displacement	roll (ψ), pitch (θ), yaw (ϕ)	degrees [°]
linear displacement	surge (x), sway (y), heave (z)	metres [m]
angular velocity	$(\dot{\psi}, \dot{\theta}, \dot{\phi})$	degrees per second [°/s]
linear velocity	$(\dot{x}, \dot{y}, \dot{z})$	meters per second [m/s]
angular acceleration	$(\ddot{\psi}, \ddot{\theta}, \ddot{\phi})$	degrees per square second [°/s^2]
linear acceleration	$(\ddot{x}, \ddot{y}, \ddot{z})$	metres per square second [m/s^2]

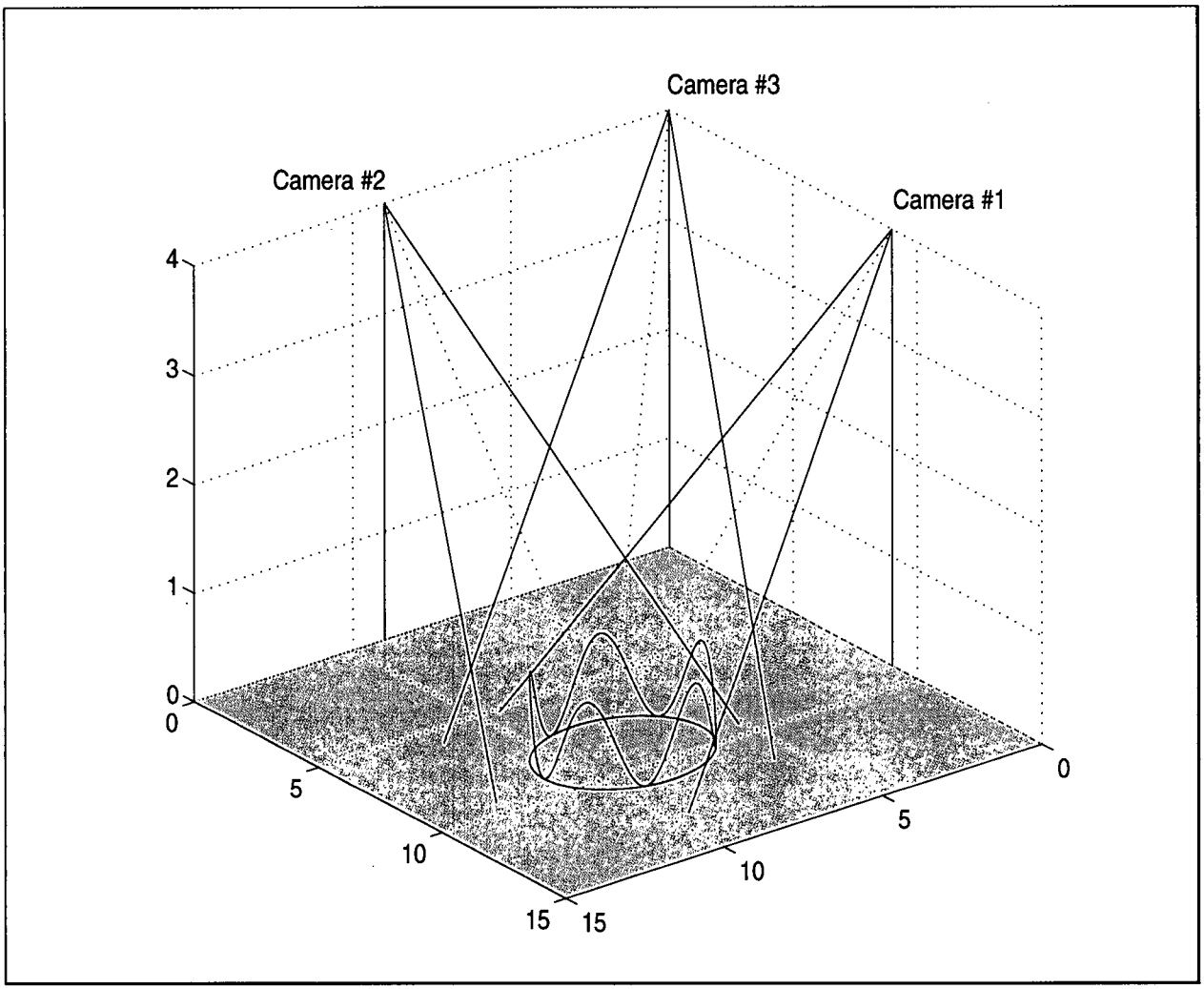


Figure 6.1: Physical Setup Represented in Simulation.

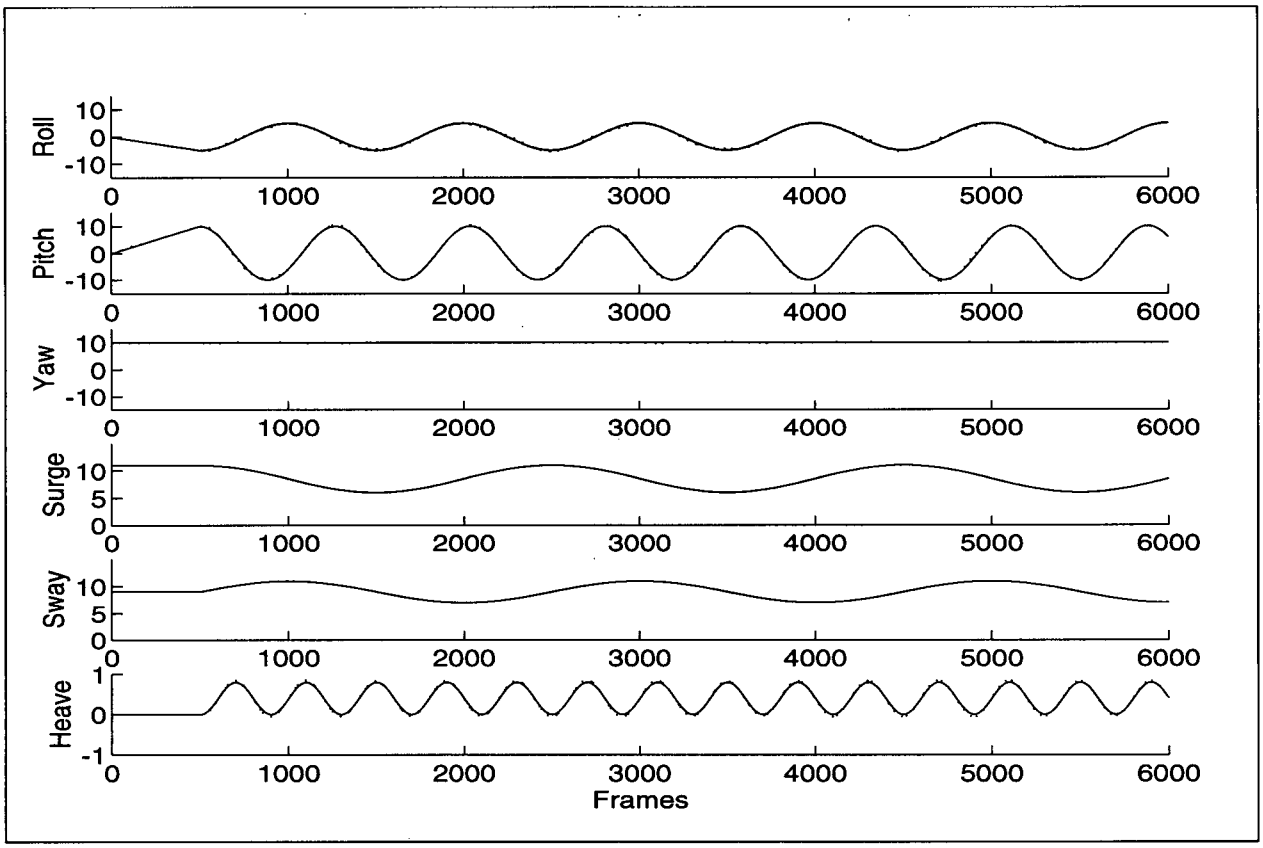


Figure 6.2: Simulation Results: Pose.

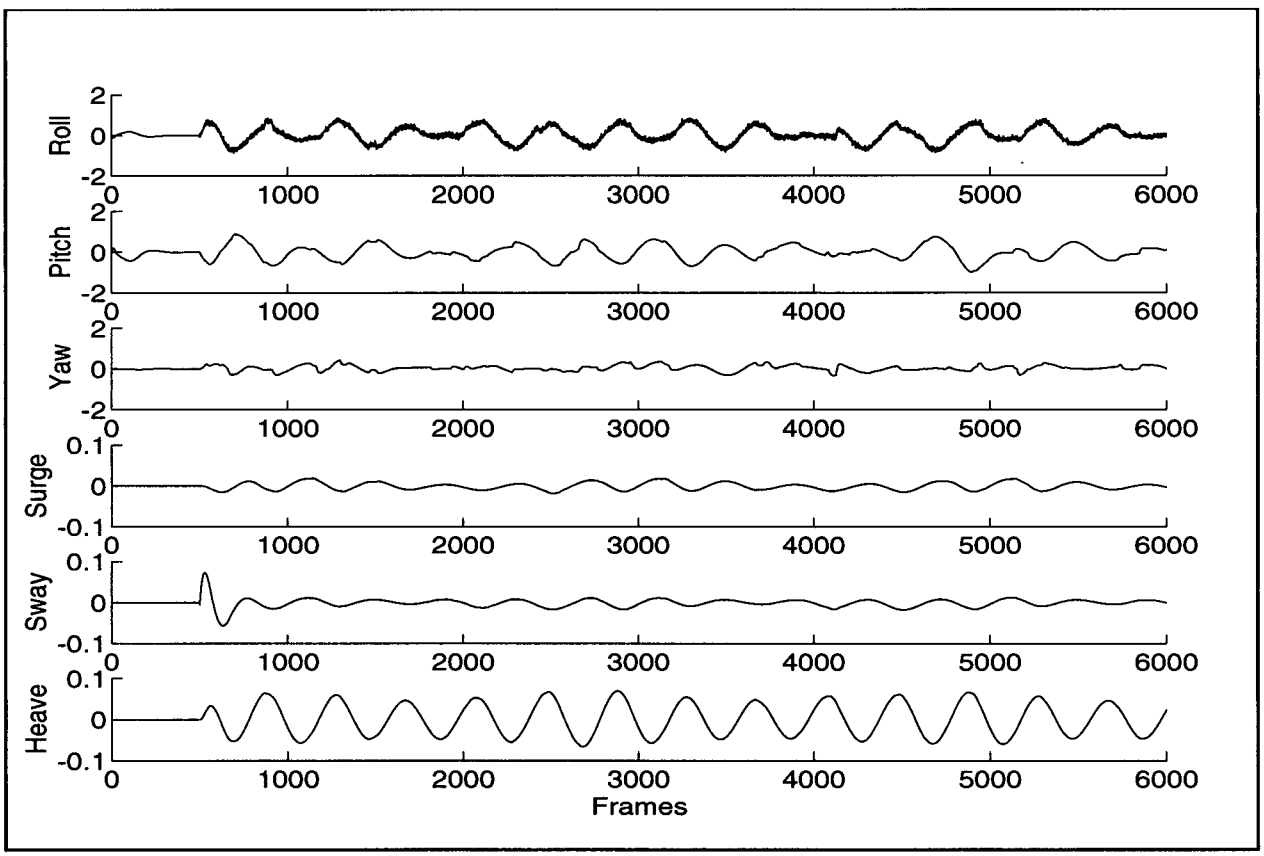


Figure 6.3: Simulation Results: Pose Error.

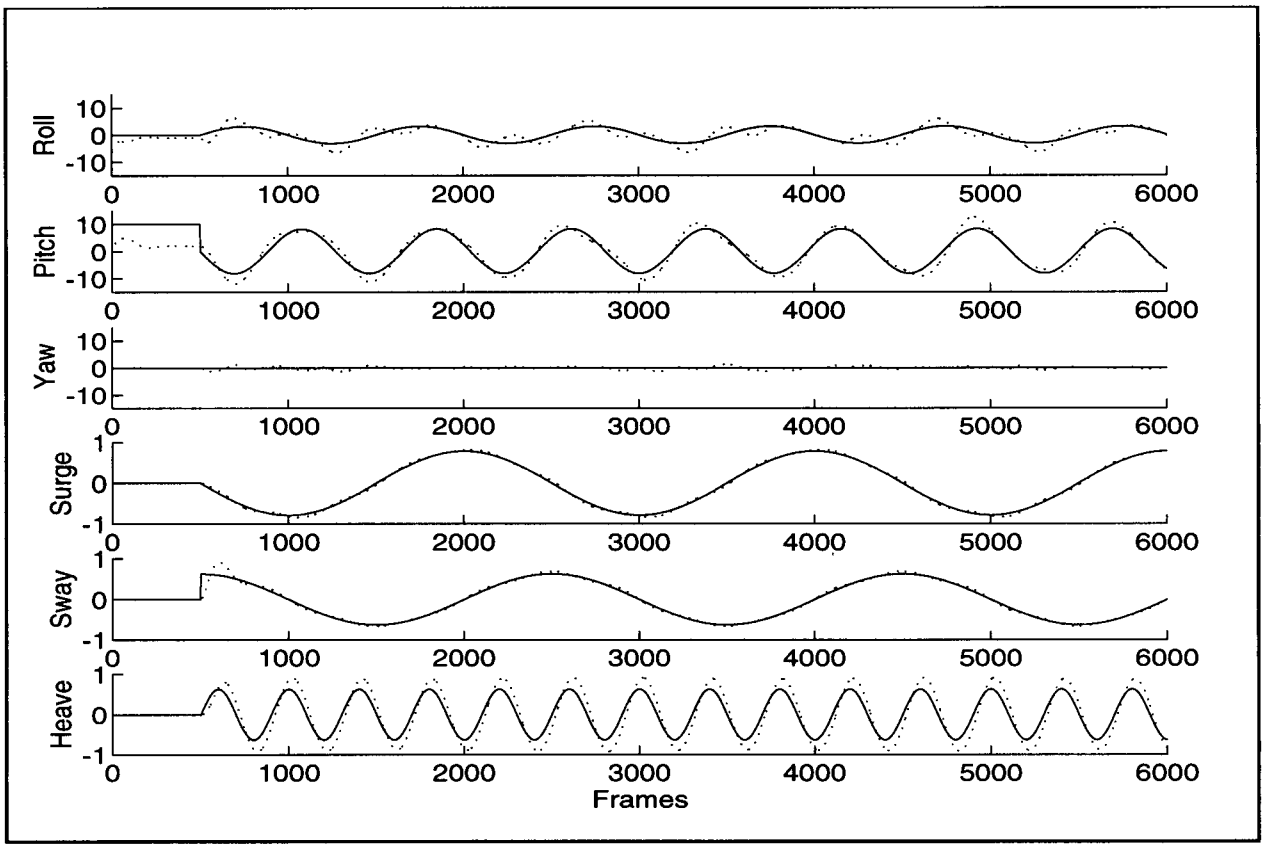


Figure 6.4: Simulation Results: Pose Velocity.

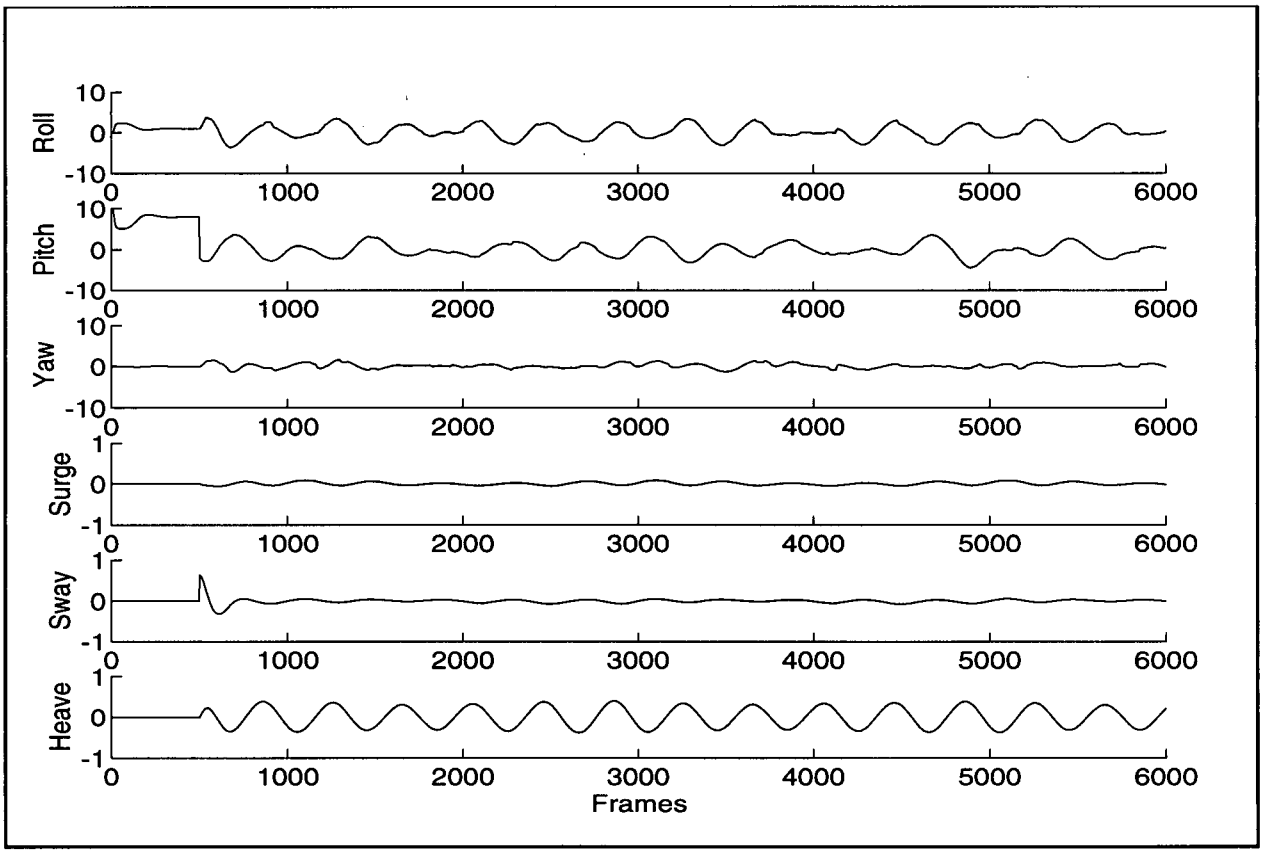


Figure 6.5: Simulation Results: Pose Velocity Error.

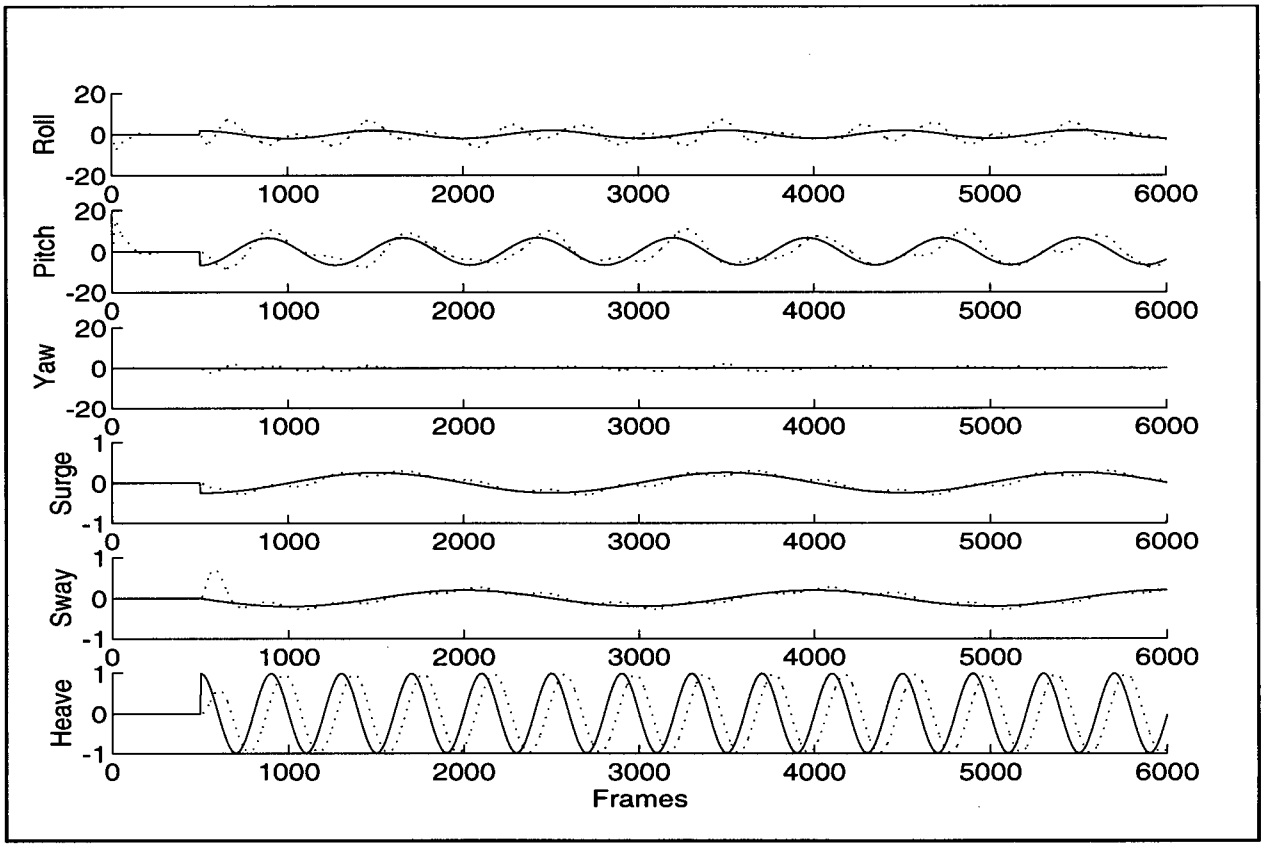


Figure 6.6: Simulation Results: Pose Acceleration.

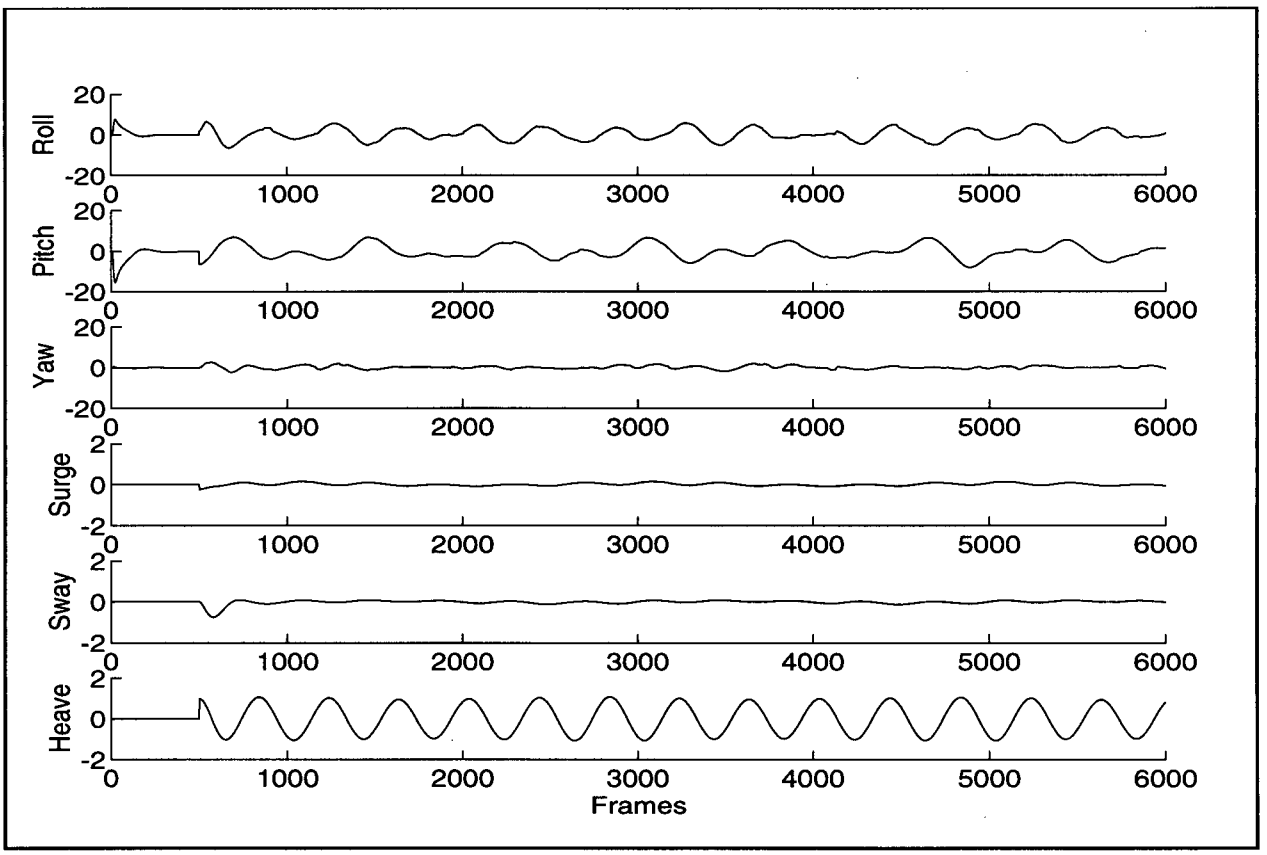


Figure 6.7: Simulation Results: Pose Acceleration Error.

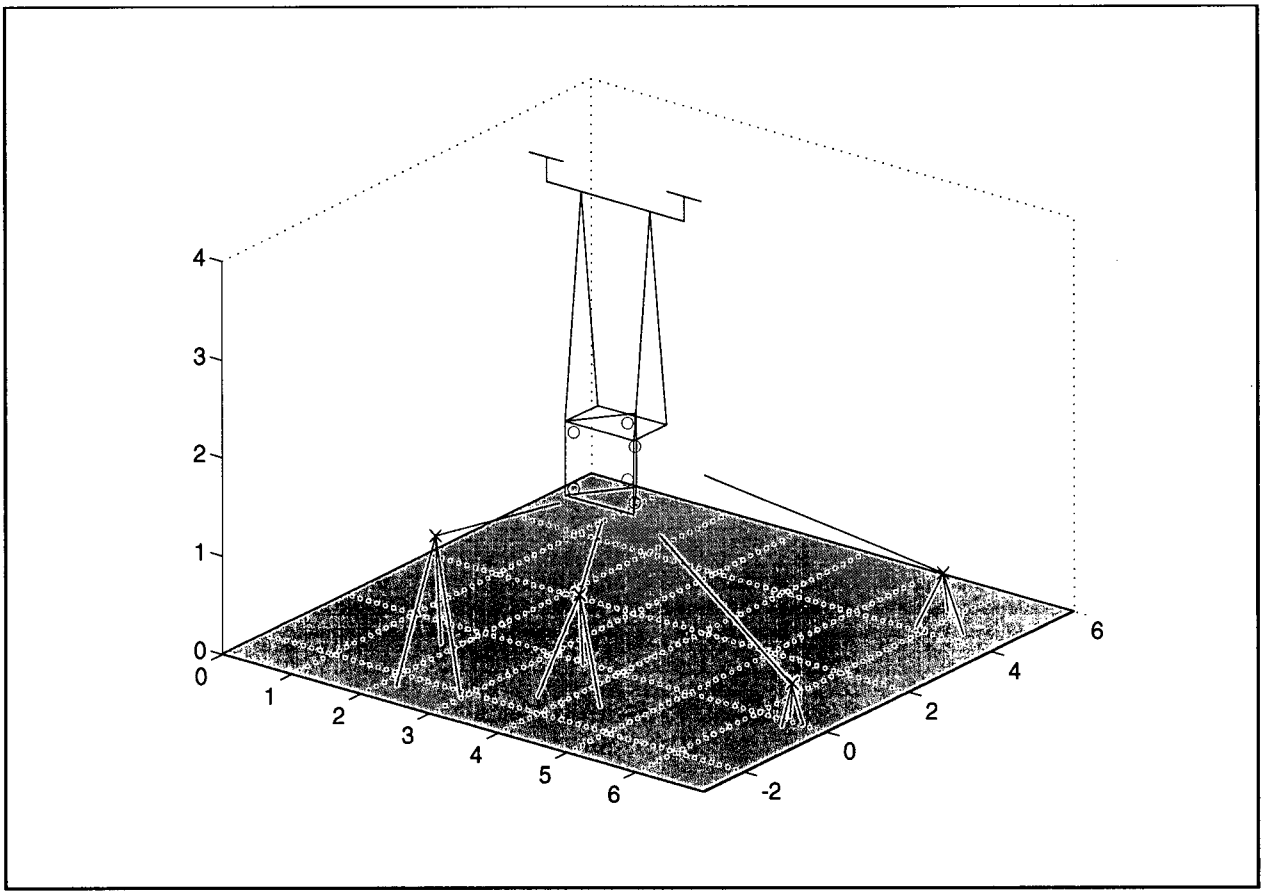


Figure 6.8: Physical Setup of Experimental Testing.

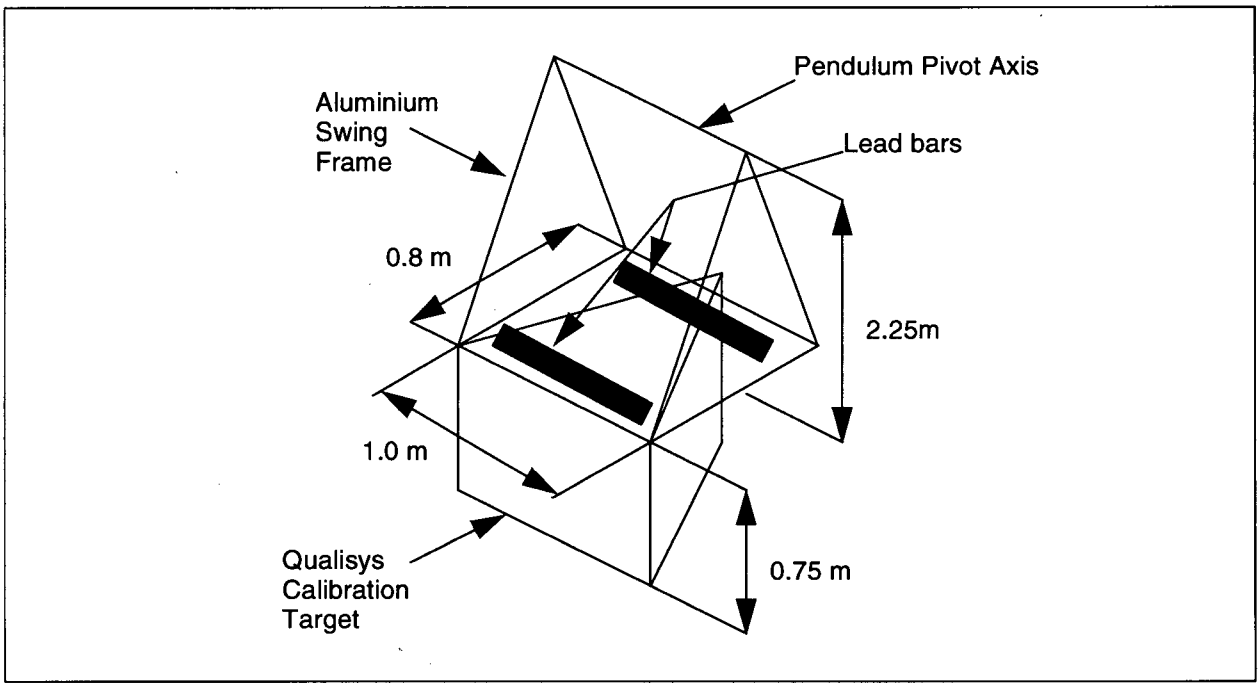


Figure 6.9: Experimental Pendulum Geometry.

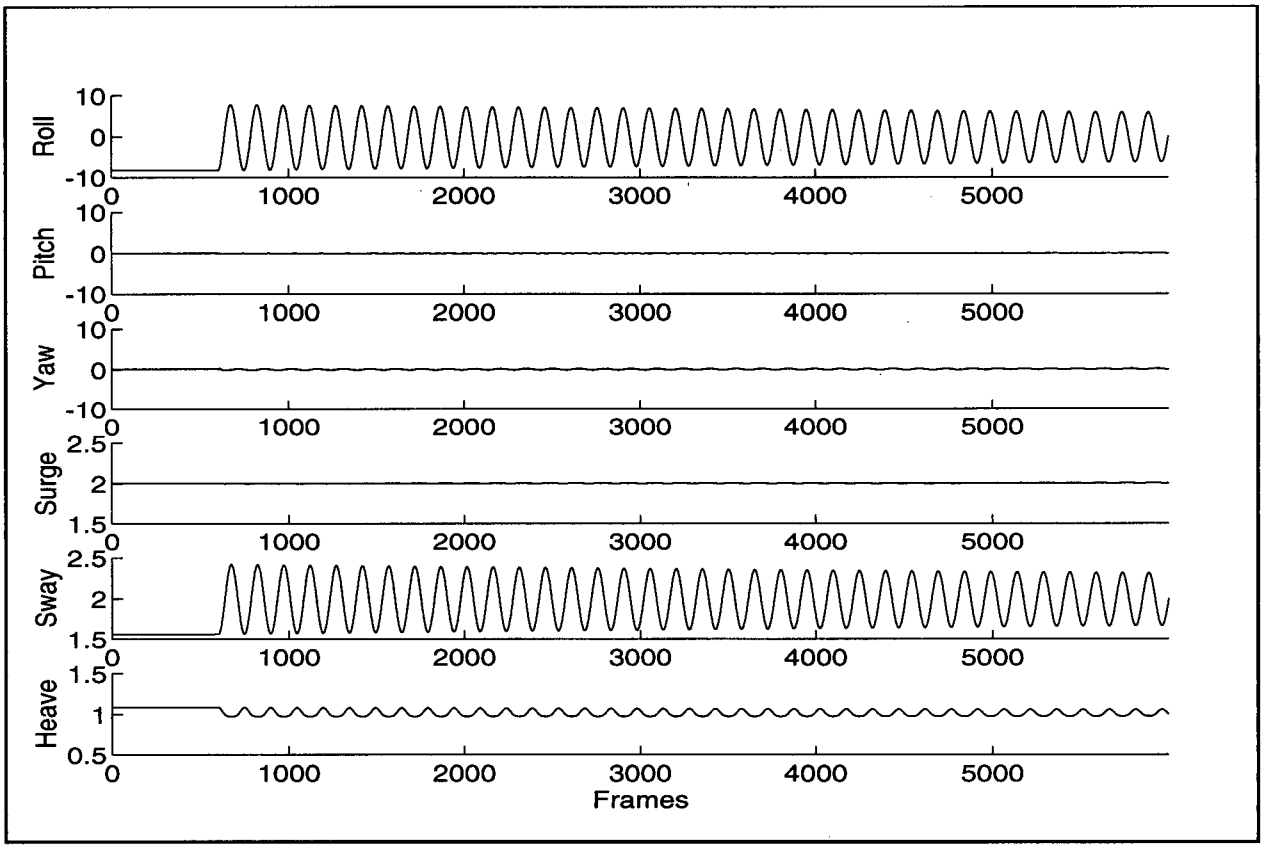


Figure 6.10: Experimental Results: Pose.

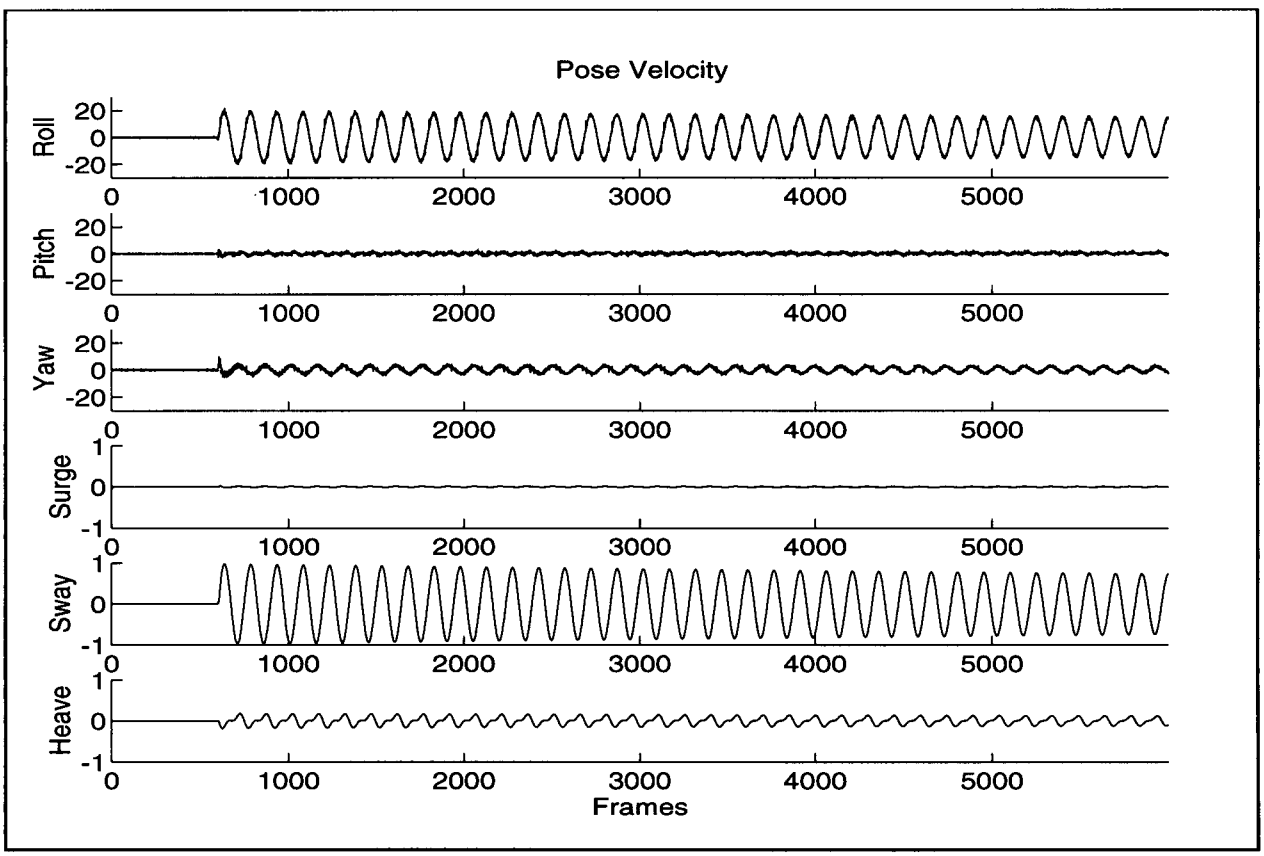


Figure 6.11: Experimental Results: Pose Velocity.

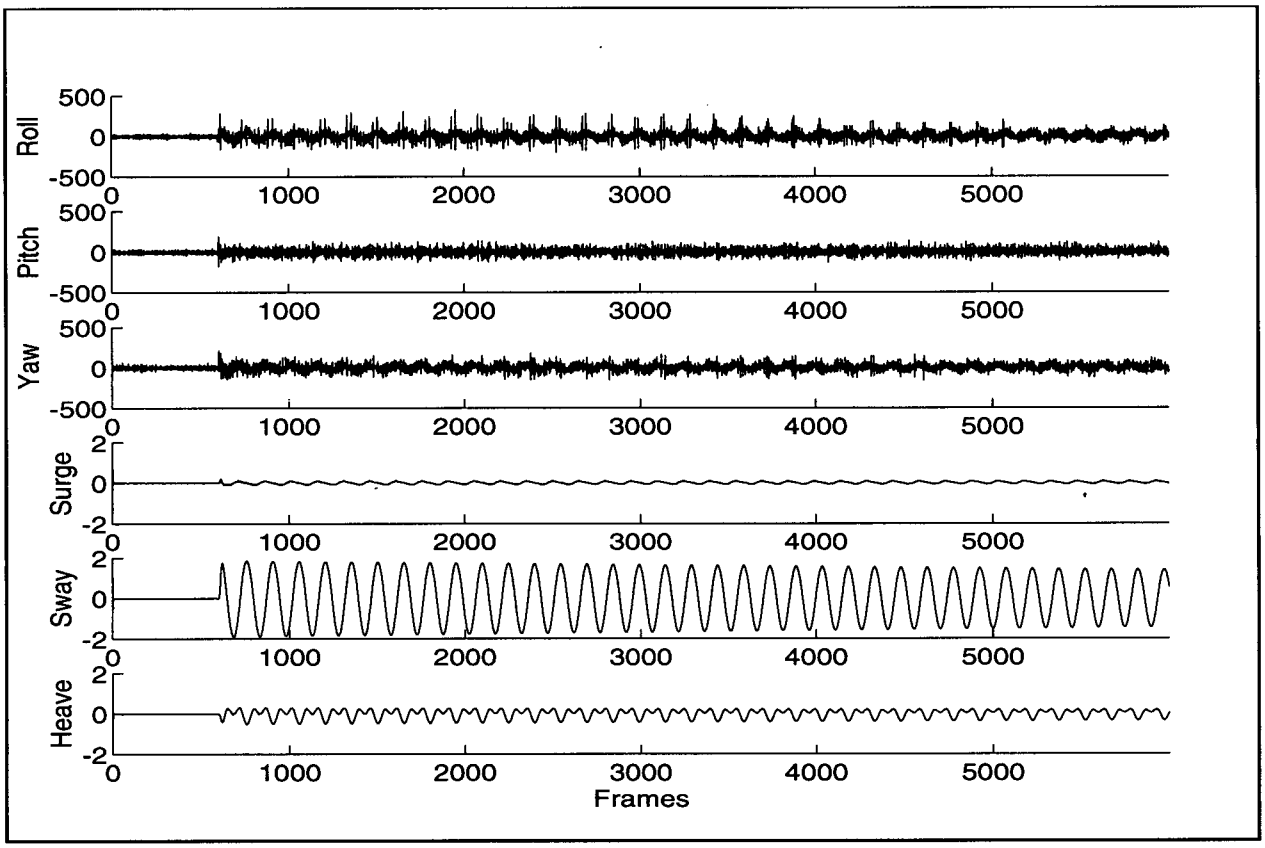


Figure 6.12: Experimental Results: Acceleration.

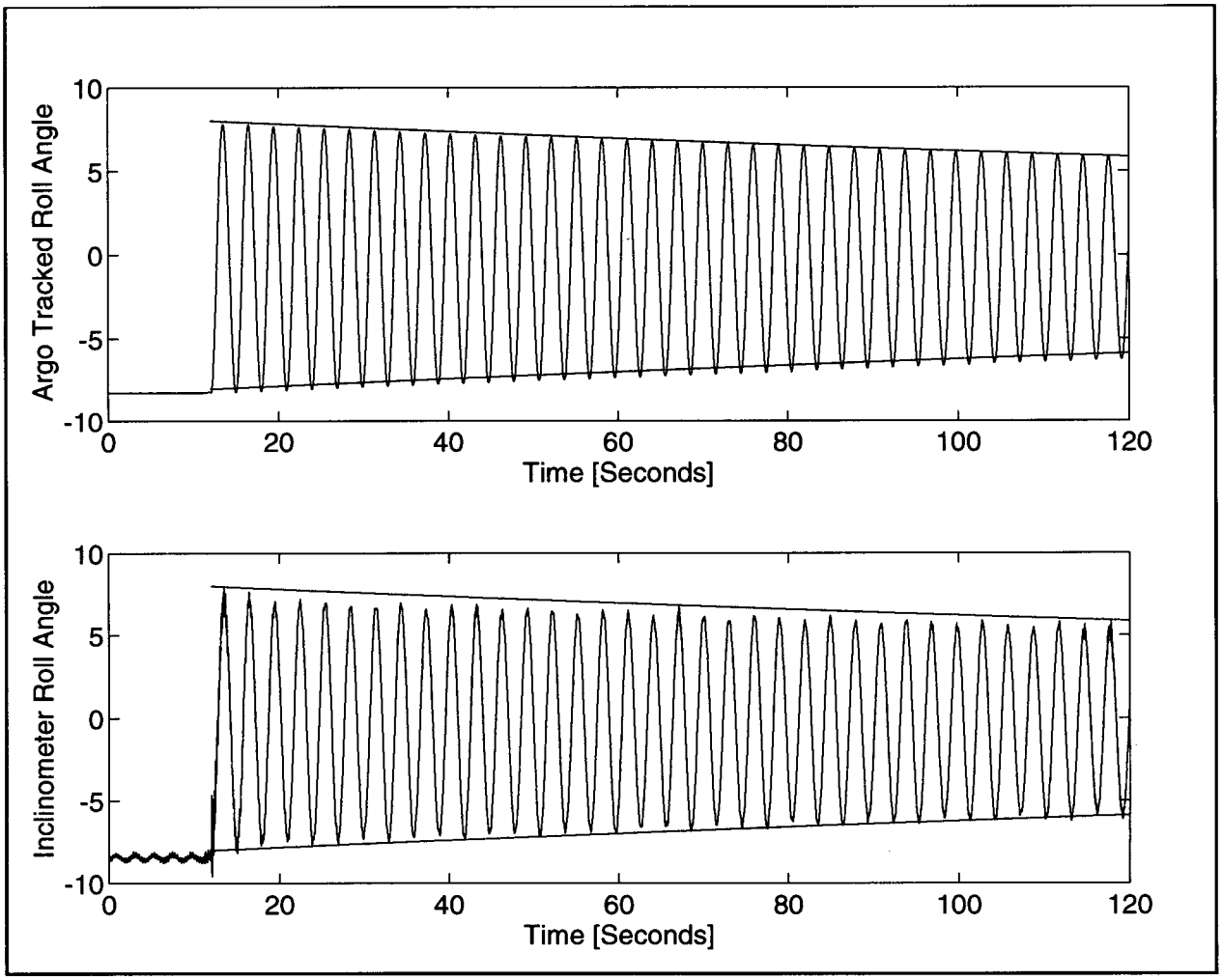


Figure 6.13: Experimental Results: Roll Angle.

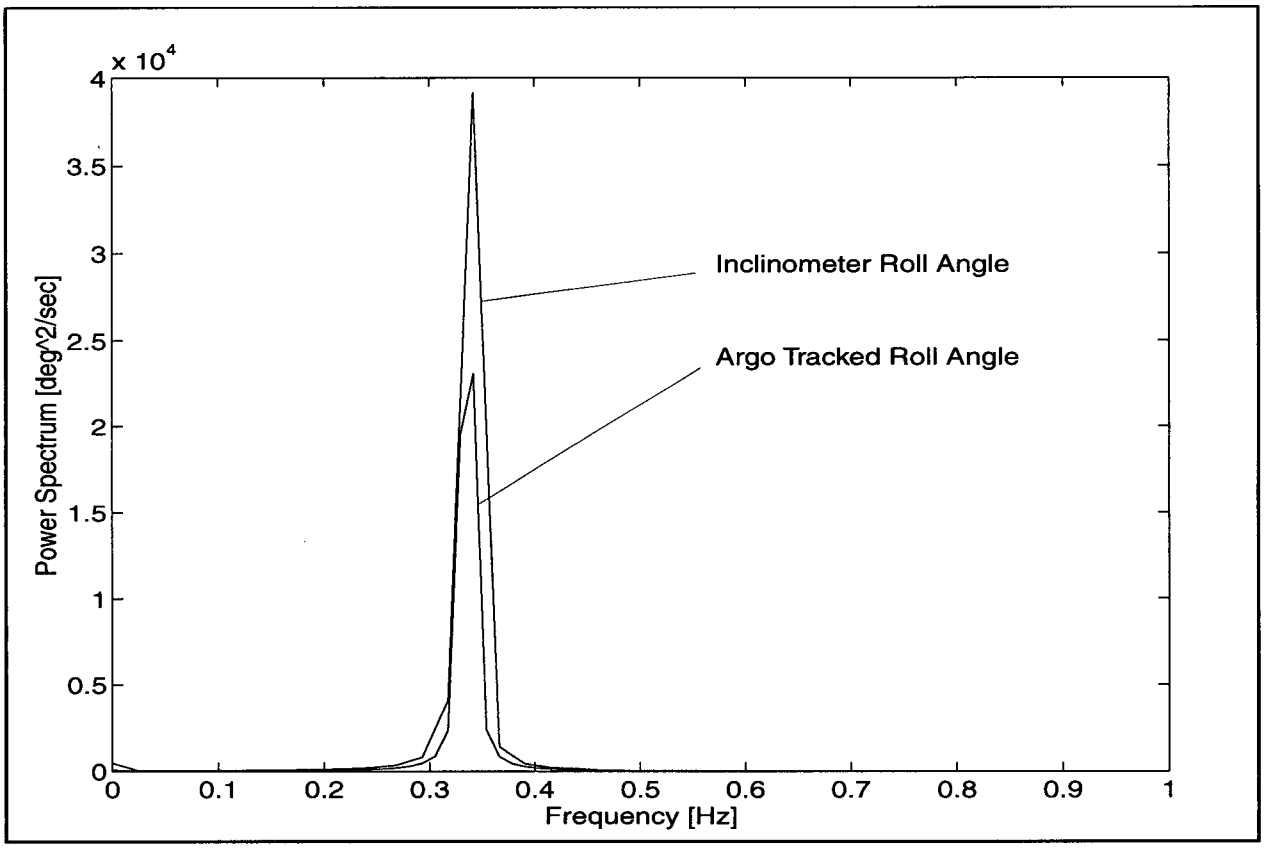


Figure 6.14: Experimental Results: Frequency Comparison of Roll Angle.

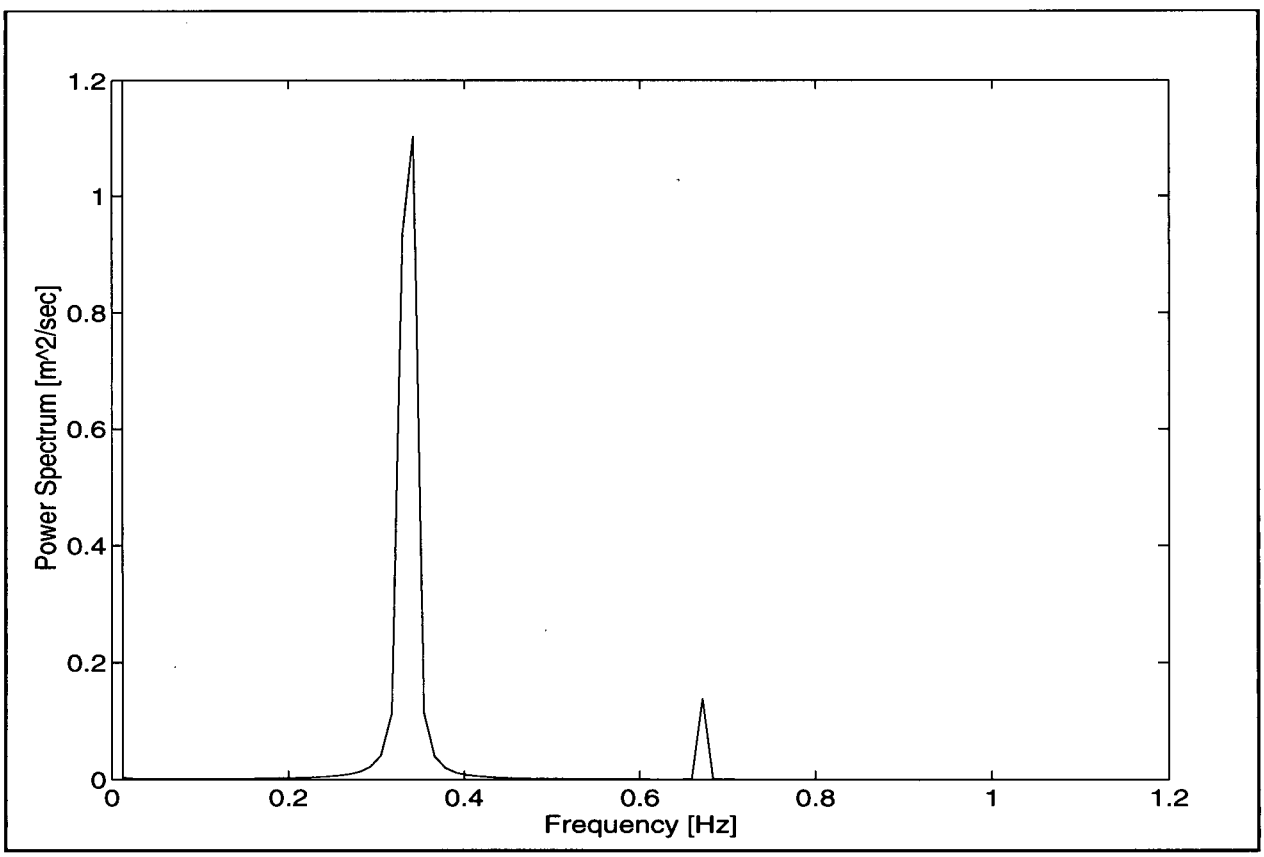


Figure 6.15: Experimental Results: Frequency of Argo Tracked Heave.

Chapter 7: Conclusions and Recommendations

7.1 Conclusions

The Argo Project has been successfully demonstrated to track the large amplitude, low frequency, rigid body motion of a target with known geometry, without constraining its motion. This tool, although developed specifically to evaluate the response motions of model ships in towing and manoeuvring tanks, is shown to be a very effective non-contact 6 DOF target trajectory measurement tool that could be applied to numerous situations.

From a review of non-contact position and motion measurement devices, machine vision was selected as the sensor for this project over other measurement systems for two reasons: (*i*) the characteristics of machine vision most closely matched the requirements of the project, and (*ii*) machine vision had been successfully implemented in similar systems already in use in towing and manoeuvring tanks facilities around the world.

This work approaches the problem of tracking the rigid body motion of a model ship, directly from image data, rather than triangulating the positions of individual markers and then solving for the pose as many other existing systems do. The pose of a target with known geometry can be solved for, from a set of nonlinear constraint equations, that use data collected from an individual or multiple cameras. This set of nonlinear constraint equations are collectively known as the observation model and are based on a 6 DOF coordinate transformation and the direct linear transformation (DLT) camera model.

An extended Kalman filter (EKF) is used to track the rigid body motion of the target. The EKF was selected because it is capable of state estimation based on indirect non-linear observation of state variables, multi-sensor data fusion, and signal noise reduction. The multi-camera-observation model is incorporated into the EKF to relate the incoming image coordinate-data to the state

variables that describe the pose of the model ship.

The new tracking algorithm resulting from this research effort was tested with computer simulation and with data gathered from physical experiments. In the numerical simulation tests the system successfully tracked a dynamic system but was limited to cyclic motions of less than 0.25 Hz because of a following phase lag between the actual and tracked trajectories. During the physical experiments the system observed the motion of a target of known geometry attached to the free end of a pendulum. The results of these tests were satisfactory as a preliminary validation, but also indicated possible sources of errors, as the tracked data did not perfectly match with the expected results. Some of the potential sources of error in the experimental test results, from both simulation and physical testing are: (*i*) calibration error, (*ii*) lack of stiffness in the test rig, (*iii*) numerical linearization errors, and (*iv*) the EKF was not optimally tunned for the system under observation.

This research effort has direct application to the marine testing industry and could be the basis of a task specific tracking system. It is believed that such a system can be assembled with off the shelf components that would be flexible in its design. With small changes external to the core tracking algorithm, the system could be reconfigured and optimized to match a variety of motion measurement tasks. The deficiencies identified in this prototype system will be dealt with in future versions of the tracking algorithm. As well, incorporating additional functionallity into the system will lead to a fully working tool that can be used in towing and manoeuvring tanks to evaluate the response motions of a model, with confidence in the results.

7.2 Future Work and Recommendations

It is the opinion of the principle researcher of this project that, although it has been demonstrated to function within the confines of controlled experiments, there is additional work required to make it a fully functioning product. Much validation testing of this program will be required before this system can be employed with confidence by researchers as a measurement tool. Some of the improvements and additions that should be considered are:

- Adding lens distortions to the observation model to improve accuracy of the estimates of the observable outputs when the target in the image moves away from the focal axis of the camera.
- Refinement of the calibration method, to make it more user friendly and more accurate, is necessary to increases the effective working volume. The current observation model and calibration method proved problematic when processing experimental data that travelled away from the centre of the working volume.
- Trajectory track recovery should be included in a future version. If the tracking algorithm loses track of the target the program should go into a search mode to re-establish the track if possible. In a self propelled model seakeeping test it is conceivable that the target could be momentarily occluded from view such that insufficient data is gathered and the EKF will loose track. Presently, once the program has lost track of the trajectory it terminates in a semi-controlled manner.
- Improving the data correlation and validation algorithm will make the tracking process more robust in handling noise in the images.
- Upgrading the camera model in SoftCam to be more representative of how a real camera generates an image of a scene is necessary for the development and testing of a more accurate observation model.

- The ability to track multiple mobile rigid bodies simultaneously is desirable when looking at the interaction of two floating bodies, such as a tug and barge experiment in a towing tank. This could be achieved by adding a second parallel EKF and modifying the pair matching process to split the input vector in two, to feed into the EKF pair.
- Tracking the pose of a moving camera with respect to a fixed set of feature points in the scene. This problem is very similar to the calibration of the external orientation of a camera.
- Modify the observation model so that the camera group can be moved relative to a fixed reference frame. This is necessary if the cameras are to be mounted to a carriage and are tracking a model ship with respect to a fixed point. This should be attacked from two fronts: odometry from the carriage control and from fixed feature points in the scene that the vision system could track its own position.
- A complicated problem that may be more academic than necessary is to track the motion of a mobile rigid body with independently moving cameras. This may have no immediate use but with increasing demands for tele-presence by organizations like NASA, the entertainment industry and the medical community, it is likely to be needed in the near future.
- The program should be ported to a compiler based programming language, most likely C, so that it can be made stand alone and can be made to operate under a real-time operating system.

Bibliography

[Aggarwal 97] J. K. Aggarwal and Q. Cai, "Human Motion Analysis: A Review"," *Proceedings of IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects*, 90-102, 1997.

[Alexander 97] Gregory H. Alexander, "An Automated Ship Model Tracking System (Manoeuvering Experiments without a Purpose-Built Facility)," *4th Canadian Marine Hydromechanics and Structures Conference*, Ottawa, ON, Canada, June 25-27, 1997.

[Allen 96] Ronald L. Allen, "Strategies for Real-Time Motion Analysis", in *Real-Time Imaging: Theory, Techniques, and Applications*, P. A. Laplante and A. D. Stoyenko (editors), IEEE Press, New York, NY, USA, 1996.

[Anderson 79] Brain D. O. Anderson and John B. Moore, "Optimal filtering," Prentice Hall, New Jersey, USA, 1979.

[Anglin 93] Carolyn Anglin, *A Functional Task Analysis and Motion Simulation for the Development of A Powered Upeer-Limb Orthosis*, Master's Thesis, Deartment of Mechanical Engineering, University of British Columbia, Canada, 1993.

[Ariel 97] Product literature and manual for Ariel Performance Analysis System, Ariel Dynamics Inc., California, USA, 1996.

[Aziz 74] Y. I. Abdel-Aziz and H. M. Karara, *Photogrammtric Potentials of Non-Metric Cameras*, Univeristy of Illinois at Urbana-Champaign, Department of Civil Engineering, March 1974.

[Bar-Shalom 88] Yaakov Bar-Shalom and Thomas E. Fortmann, *Tracking and Data Association*, Academic Press Inc., Florida, USA, 1988.

[Bar-Shalom 93] Yaakov Bar-Shalom and Xiao-Rong Li, *Estimation and Tracking: Principles, Techniques, and Software*, Artech House Inc., Boston, MA, 1993.

[Blachut 89] Teodor. J. Blachut and Rudolf Burkhardt, *Historical Development of Photogrametric Methods and Instruments*, Ameriacan Society for Photogrammetry and Remote Sensing, 1989.

[Blackburn 96] Dave Blackburn, "Wired Athletes," IRIS Universe, No. 36, Silicon Graphics Inc., 1996.

[Beer 84] Ferdinand P. Beer and E. Russel Johnston Jr., *Vector Mechanics for Engineers Statics and Dynamics 4th Ed.*, McGraw-Hill Book Company, New York, 1984.

[BNS 94] Baltic News Service, "Estonia Passenger Ferry Sinks, More Than 750 Feared Dead", 12:00, September 28, 1994.

[Borenstien 96] J. Borenstein, H. R. Everett and L. Feng, *"Where am I?" Sensors and Methods for Mobile Robot Positioning*, containted on Mobile Robotics Lab 10-year Anniversary Research Report CD, University of Michigan, 1996.

[Bozic 94] S. M. Bozic, *Digital and Kalman Filtering 2nd Ed.*, Wiley, New York, 1994.

[Charnwood 97] Product literature regarding CODA mpx 30, Charnwood Dynamics Ltd., Leicestershire, UK, 1997

[Chui 91] C. K. Chui and G. Chen, *Kalman Filtering With Real-Time Applications 2nd Ed.*, Springer-Verlag, New York, 1991.

[CIDTEC 97] Product literature for CIDTEC cameras and sensors, CIDTEC, New York, USA, 1997.

[Columbus 97] Product literature from Columbus Instruments for thier VIDEOMEX-ONE, VIDEOMEX-V and VIDEOMEX-X products, Columbus Instruments International Corporation, Columbus, Ohio, USA, 1997.

[Cook 81] David A. Cook, *A History of Narrative Film*, W.W. Norton & Company, Inc., New York, USA, 1981.

[Coombes 96] Jennifer Coombes, "Seeing in Space Shuttle Astronauts Get a Better View with ASVS," article in QNX News, Vol 10, No. 3 & 4, 1996.

[Dainis] Andrew Dainis and Maris Juberts, "Accurate Remote Measuremnet of Robot Trajectory Motion," authors from: Industiral Systems Division Center for Manufacturing Engineeirng, National Bureau of Standards, Gaithersburg, MD 20899, source and date of publication unknown.

[Dalsa 96] Dalsa 1996-1997 Databook for CCD Image Capture Technology, Dalsa Inc., Ontario, Canada, 1996.

[deSilva 89] Clarence deSilva, *Control Sensors and Actuators*, Prentice-Hall, New Jersey, 1989.

[Delaney 97] Ben Delaney, "The Science of Motion Capture," Innovation^3, No. 40, 52-56, Silicon Graphics Inc., summer 1997.

[Duda 73] Richard O. Duda and Peter E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, USA, 1973.

[Dynascope 96] Product literature for Dyanscope: Optical System for Motion Analysis, Sirehna, Nantes, France, 1996.

[Edgerton 87] Harold E. Edgerton, *Electonic Flash, Stobe*, 3rd Ed., MIT Press, Mass., USA, April, 1987.

[El-Hakim 92] Sabry F. El-Hakim, Nicolino J. Pizzi and David Westmore, "The VCM Automated 3-D Measurement System - Theory, Application and Performance Evaluation," in *Applications of Artificial Intelligence X: Machine Vision and Robotics*, SPIE Vol. 1708, 460-482, 1992.

[Fang 83] J. Q. Fang and T. S. Huang, "Estimating 3-D Movement of a Rigid Object: Experimental Results," Proceedings of the 8th International Joint Conferences on Artificial Intelligence, Karlsruhe, West Germany, Vol. 2, 1035-1037, August 8-12, 1983.

[Faugeras 93] Oliver Faugeras, *Three-Dimensional Computer Vision A Geometric Viewpoint*, MIT Press, Massachusetts, USA, 1993.

[Fu 87] K. S. Fu, R. C. Gonzalez & C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill, New York, NY, USA, 1987.

[Gieck 90] Kurt Gieck and Reiner Gieck, *Engineering Formulas 6th Ed.*, McGraw-Hill, West Germany, 1990.

[Golub 89] Gene H. Golub and Charles F. Van Loan, *Matrix Computations 2nd Ed.*, Johns Hopkins University Press, Baltimore, MD, USA, 1989.

[Gosine 96] Ray Gosine, Lecture Notes 1996, Memorial University, Faculty of Engineering, 1996.

[Gospodnetic 92] Slobodan Gospodnetic, John Phillips and Michal D. Miles, *Computer Vision System for Motion Measurement of Small Floating Bodies*, Dominis Engineering Ltd. Ontario, Canada, 1992.

[Greaves] John O. B. Greaves, "Interlacing Error in Video Systems: Hardware and Software Issues", Motion Analysis Corporation, publication data unknown.

[Grewal 93] Mohinder S. Grewal and Angus P. Andrews, *Kalman Filtering Theory and Practice*, Prentice Hall, New Jersey, USA, 1993.

[Hallan 83] John Hallan, "Resolving Observer Motion by Object Tracking," Proceedings of the 8th International Joint Conferences on Artificial Intelligence, Karlsruhe, West Germany, Vol. 2, pp. 792-798, August 8-12, 1983.

[Harris 96] Gerald F. Harris and Peter A. Smith eds., *Human Motion Analysis: Current Applications and Future Directions*, IEEE Press, New York, USA, 1996.

[Horn 86] Berthold Klaus Paul Horn, *Robot Vision*, MIT Press and McGraw-Hill, USA, 1986.

[Hosie 95] Robin Hosie, Predicting Ball Trajectories in Uncontrolled Environments, honours Undergraduate Thesis, Curtin University of Technology, Perth, Western Australia, January 1995.

[IMD 5] *IMD News #5*, National Research Council of Canada, Institute for Marine Dynamics, publication date unknown.

[IMD 6] *IMD News #6*, National Research Council of Canada, Institute for Marine Dynamics, publication date unknown.

[Johnson 90] Howard Johnson, "The Mechanical Engineer and the Transsition to Image Analysis," Advanced Imaging, NY, USA, November, 1990.

[Kalman 60] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," ASME Journal of Basic Engineering, Vol. 82, Series D Number 1, pp 35-45, 1960.

[Karara 89] H. M. Karara ed., *Non-Topographic Photogrammetry 2nd Ed.*, American Society of Photogrammetry and Remote Sensing, 1989.

[Kinetic 97] Product literature and personal communications regarding Eagle Eye, Kinetic Sciences Inc., Vancouver, British Columbia, Canada, 1997.

[Krishnan 92] Radha Krishnan, H. J. Sommer III and Peter D. Spidaliere, "Monocular Pose of a Rigid Body USing Point Landmarks," CVGIP Image Understanding, Vol. 55, No. 3, May, pp 307-316, Academic Press, 1992.

[Lavic 93] Product literature for Lavic: Three dimensional motion real-time analysis system. (Biomechanics video motion capture), Emtec Co. Ltd., Tokyo, Japan, 1993.

[Lin 86] Zse-Cherng Lin, Thomas S. Huang, Steven D. Blostein, Hua Lee, and E. A. Margerum, "Motion Estimation from 3-D Point Sets With and Without Correspondences," *Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, USA, 194-201, June 22-26, 1986.

[Lowe 92] David G. Lowe, "Robust Model-based Motion Tracking Through the Integration of Search and Estimation," *International Journal of Computer Vision*, 8:2, 113-122, 1992.

[Mackay 96] David Mackay, "Underwater Cameras - SIT or CCD?", *Sea Technology*, Volume 37, No. 9, 35-38, Compass Publications, September, 1996.

[Manku] Gurmeet Singh Manku, Pankaj Jain, Amit Aggarwal, Lalit Kumar, and Subhashis Banerjee, "Object Tracking using Affine Structure for Point Correspondences," paper published on internet, Dept. of Computer Sciences and Engineering, Indian Institute of Technology, New Delhi, India, date of publication unknown.

[MATLAB 92] *MATLAB Reference Guide Ver 4.0*, The MathWorks, Inc., USA, 1992.

[MARINTEK 97] Information from MARINTEK world wide web home page.
http://www.marintek.sintef.no/
document relating to OPTOPOS
http://brage.marintek.sintef.no/marintek_eks/innovati.html
document relating to Selspot
http://brage.marintek.sintef.no/marintek_eks/hydrolab.html

[Marzan 76] G. T. Marzan and H. M. Karara, *Rational Design for Close-Range Photogrammtry*, Univeristy of Illinois at Urbana-Champaign, Department of Civil Engineering, January 1976.

[McGee 85] Leaonard A. Mcgee and Stanley F. Schmidt, "Discovery of the Kalman Filter as a Practical Tool For Aerospace and Industry," NASA Technical Memorandum 86847, NASA Ames Research Center, November 1985.

[Miller 80] Norman R. Miller, Robert Shapiro, and Thomas McLaughlin, "A Technique for Obtaining Spatial Kinematic Parameters of Segments of Biomechanical Systems from Cinematographic Data," *Journal of Biomechanics*, Vol. 13, 535-547, 1980.

[Motion 96] Product literature for Motion Analysis (Biomechanics video motion capture), Motion Analysis Corp., California, USA, 1996.

[Motion 97] Personal correspondance with Motion Analysis Corporation regarding use of their product by U. S. Army Corps of Engineers, 1997.

[Mulder 94] Alex Mulder, "Human Movement Tracking Technology," NSERC Hand Centered Studies of Human Movement Project, Technical Report 94-1, Simon Fraser University, B.C., Canada, 1994.

[Nalwa 93] Vishvjit S. Nalwa, *A Guided Tour of Computer Vision*, Addison-Wesley Publishing Company, USA, 1993.

[NDI 97] Product literature for Optotrak: 3D Motion Measurement System, Northern Digital Inc., Ontariio, Canada, 1997.

[Neptec 96] Presentation material for Advanced Space Vision System (ASVS), Neptec Design Group Ltd., Ontario, Canada, 1996.

[Nesi 96] P. Nesi, "Real-Time Motion Estimation", in *Real-Time Imaging: Theory, Techniques, and Applications*, P. A. Laplante and A. D. Stoyenko (editors), IEEE Press, New York, NY, USA, 1996.

[On-Trak 97] Product literature for SiTek Electro Optics Position Sensing Detectors, On-Trak Photonics Inc., California, USA, 1997.

[Paul 81] Richard P. Paul, *Robot Manipulators: Mathematics, Programming and Control,* MIT Press, Cambridge, Mass., USA, 1981.

[Peak 96] Product literature for Peak Motous System (Biomechanics video motion capture), Peak Performance Technologies Inc., Colorado, USA, 1996.

[Qualisys 92] Qualisys AB, *Qualisys Position Sensor Technical Reference Manual*, Partille Sweden: Qualisys, 1992.

[Qualisys 96] Product literature for Qualisys Position Sensor System & Multi Traks, Qualisys AB, Sweden, and Adaptive Optics of Ameraica (AOA), Hamilton Standard, Mass., USA, 1996.

[Rediers] Benny Rediers and Randall J. Allemang, "Video Sensing in Data Acquisition Systems," Structural Dynamics Research Laboratory, Department of Mechanical and Industrial Engineering, University of Cincinnati, supplied by Motion Analisys, publication data unknown.

[Romilly 94] D. P. Romilly, C. Anglin, R. G. Gosine, C. Hersler and S. U. Raschke, "A Functinoal Task Analysis and Motion Simulation for Development of a Powered Upper-Limb Orthosis", IEEE Transactions on Rehabilitation Engineering, Vol. 2, No. 3, September 1994.

[Safaee-Rad 87] Reza Safaee-Rad, *Functional Human Arm Motion Study with a New 3-D Measurement System*, Master's Thesis, Department of Electrical Engineering, University of Manitoba, Canada, 1987.

[Sasazawa 91] Shigeo Sasazawa, "Non-Contact, Real-Time, 3-Dimensional-Motion Analyzing System by Color Vision," Journal of Robotics and Mechatronics Vol. 3, No. 3, 207-210, 1991.

[Schmidt 70] S. F. Schmidt, "Comptational techniques in Kalman filtering," in *Theory and Applications of Kalman Filtering*, AGARDograph 139, NATO Advisory Group for Aerosspace Research and Development, London, February 1970.

[Selspot 96] Product literature for Selspot: 3D Motion Analysis, Selspot AB, Sweden, 1996.

[Sezan 93] M. I. Sezan and R. L. Lagendijk (editors), *Motion Analysis and Image Sequence Processing*, Kluwer Academic Publishers, Boston, USA, 1993.

[Shan 95] S. Shan and J. K. Aggarwal, "Autonomous Robot Navigation Using Fish-Eye Lenses," in *Proceedings of 1995 Third International Conference in Computer Science*, Hong Kong, December, 1995.

[Shapiro 77] Robert Shapiro, "Direct Linear Transformation Method for Three-Dimensional Cinematography," *The Research Quarterly*, Vol 49, No. 2, 197-205, 1977.

[Simon 90] Christophe Simon, *Application of the Kalman Filter to Iceberg Motion Forecasting*, University of British Columbia, Dept. of Mechanical Engineering, Master's Thesis, July 1990.

[Sirehna 97] Product and services literature for Dynascope, Sirehna, Nantes, France, 1997.

[Stremler 82] Ferrel Stremler, *Introduction to Communications Systems 2nd Ed.*, Addision-Weley Publishing Company, Massachusetts, USA, 1982.

[Sullivan 93] Micheal Sullivan, "Design of a Real-Time Motion Tracking System," *1993 Canadian Conference on Electrical and Computer Engineering*, Vancouver, BC, Canada, September 14-17, 1993.

[Sullivan 97] Micheal Sullivan, "Optical Tracking at the Institute for Marine Dynamics," *4th Canadian Marine Hydromechanics and Structures Conference*, Ottawa, ON, Canada, June 25-27, 1997.

[Sullivan PC] Peersonal correspondence between author and Micheal Sullivan of the National Research Council of Canada, Institute for Marine Dynamics, 1994-1997.

[Tau 96] Product literature from Tau Corporation (visual tracking systems), California, USA, 1996.

[Trimble 97] Product literature regarding GPS and DGPS technology, Trimble Navigation Ltd., Sunnyvale, California, USA, 1997.

[Tsai 85] Roger Y. Tsai, "A Versatile Camera Calibration Technique for High Accuracy 3D MAchine Vision Metrology using Off-the-Shelf TV Cameras and Lenses," Research Report RC 11413, IBM Research Division, San Jose, California, USA, October 1, 1985.

[Veillon 96] A. Veillon, J. M. Aillaud and P. Brunet, "Submarine Depth Control Under Waves and Experimental Approach," *International Symposium Warship '96, Naval Submarines 5, "The Total Weapon System"*, London, UK, June 12-14, 1996.

[Vicon 96] Product literature for Vicon Motion Systems (Biomechanics video motion capture), Oxford Metrics, Oxford, England, 1996.

[Walton 86] James S. Walton, "The Accuracy and Precsion of a Video-Based Motion Analysis System," *Proceedings of the 30th International Technical Symposium on Optical and Optpelectronic Applied Sciences and Engineering*, Volume 693: "High-Speed Photography, Videography and Photonics IV," San Diego, CA., August 17-22, 1986.

[Walton] James S. Walton, "Expertvision: A Video-Based Non-Contact System For Motion Measurement," Motion Analysis Corporation, publication date unknown.

[Wang 91] Jiang Wang, *Optimal Estimation of 3D Relative Position and Orientation for Robot Control*, master's thesis, University of Waterloo, Canada, 1991.

[Wang 92] Jiang Wang and William J. Wilson, "3D Relative Position and Orientation Estimation Using Kalman Filter For Robot Control," *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, pp. 2638-2645, 1992.

[Welch 97] Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter," paper published on the internet, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175, September, 1997.

[Weng 92] Juyang Weng, Paul Cohen and Nicolas Rebibo, "Motion And Structure Estimation from Stereo Image Sequences," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, 362 - 282, June, 1992

[Weng 92a] Juyang Weng, Paul Cohen and Marc Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 10, 965-980, October, 1992.

[Wilson 93] Willian J. Wilson, "Visual Servo Control of Robots Using Klaman Fitler Estimates of Robot Pose Relative to Work-Pieces," In K. Hashimoto, editor, *Visual Servoing*, World Scinetific Publishing Co. Pte. Ltd., Singapore, 71-104, 1993.

[Wu 88] J. J. Wu, R. E. Rink, T. M. Caelli, and V. G. Gourishankar, "Recovery of the 3-D Location and Motion of a Rigid Object Through Camera Image (An Extended Kalman Filter Approach)," *International Journal of Computer Vision*, 3, 373-394, 1988.

[Wysner 83] Shirley Worth Wysner, "Tracking high-speed movement in three dimensions: As eas as (X,Y, Z)," *Search*, Vol. 18, No. 3, General Motors Research Laboratories, July-August, 1983.

[Yaman 97] Product literature from Yaman Ltd., Tokyo, Japan, 1997.

[Yuan 89] Joseph S.-C. Yuan, "A General Photogrammetric Method for Determining Object Position and Orientation," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 2, April 1989.

[Zha 95] Hongbin Zha, Toyoshi Onitsuka and Tadashi Nagata, "Self-Organization Based Visuo-Motor Coordination for a Real Camera and Manipulator System," *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 4, Vancouver, B.C., Oct. 22-25, 1995.

[Zhang 92] Zhengyou Zhang and Oliver Faugeras, *3D Dynamic Scene Analysis: A Stereo Based Approach*, Springer-Verlag, Berlin, Germany, 1992.

Appendix A: Motion Capture Systems

This appendix contains a summary of some of the motion capture-analysis research projects, as well as many of the commercially available systems. The varied nomenclature indicates the wide variety of methods, equipment and applications of this technology.

A.1 Abbreviations

General

N/A	Not Applicable	NP	Not Published

Marker Types

CAIR	Cyclic active IR LEDs	HILP	High Intensity Light Points
ED	Edge detection	EC&CT	Edges, Corners and Circular targets
HCBW	High Contrast Black and White Markers	HCCM	High Contrast Colour Markers
RRS	Retroreflective Spheres and Hemispheres	SFSI	Substraction of fixed scene from image

Tracking Method

EKF	Extended Kalman Filter	MBT	Model Based Tracking
MKF	Modified Kalman Filter	NN	Neural Network
NM	No motion tracking, still scene only	PO	Track position only not vel. or accel.
PPE	Previous position as estimate	PD	Differentiation of pose trajectory

Camera Model

DLT	Direct Linear Transformation	PPP	Plane Perspective Projection
CL	Colinear Equations	NLE	Nonlinear Equations

Camera Format

CCD	Charge Coupled Device Camera (B&W)	CCD-clr	Colour CCD Camera
CCD-IR	Black and white CCD camera fitted with an near IR filter	3LCCD	3 Linear Array Charge Coupled Devices in a single camera
DAC	Diode Array Camera	F	Film Cine-Camera
PSD	Position Sensing Device	CID	Charge Induction Device

Lighting

NOL	Normal overhead lighting	IRS	Infra-red strobe

Primary Uses of System

HMA	Human motion analysis	Ob	Observation of a dynamics system
PM	Tracking particle motion	RC	Robot calibration
RVS	Robot visual servoing	TMS	Tracking model ships or floating bodies
DP	Dynamic positioning of submarine		

Host Platform

PC	IBM 100% compatible personal computer	Mac	Apple MacIntosh personal computer
SGI	Silicon Graphics workstation	Sun	Sun workstation
FG	Frame grabber	PP	Proprietary processor

A.2 Video Based Motion Capture Systems Comparision Tables

Table A.1: Research Video Based Motion Capture Systems

Publication Reference	Number of Cameraos	Stereo Triangulation	Feature Points/ Marker Type	Tracking Method	DOF Rigid Body Motion	Prinary Uses	Image Format	Camera Model	Commercial Systems
Alexander	1		HILP	PP	3	TMS	CCD	NP	
Anglin	2	x	HCBW	PP	N/A	HMA	CCD	DLT	
*Argo Project	1-m	**	RR	EKF	6	TMS	CCD	DLT	Qualisys
Dainis		x	CAIR	PO	6	RC	PSD	DLT	
El-Hakim		x	EC&CT	PD	N/A	Ob	CCD	CL	
Fang	1		ED	PD	6	Ob	CID	NLE	
Gospodnetic	2	x	HCBW	NP	6	TMS	CCD	NP	
Hosie	1		SFSI	EKF	3	PM	CCD	PPP	
Krishnan	1		NP	NM	6	DP	NP	PPP	
Lowe	1		ED	MBT	6	Ob	CCD	PPP	
Manku	1		ED	MKF	2	Ob	CCD	NP	
Miller	2	x	HCBW	PD	6	Ob	F	DLT	
Shapiro	2	x	HCBW	PD	3	PM	F	DLT	
Sullivan	2	x	RR	PD	6	TMS	CCD	PPP	Qualisys
Veillon	2	x	RR	NP	6	TMS	CCD	NP	
*Wilson,Wang	1		Hole	EKF	6	RVS	DAC	PPP	
*Wu	1		HCBW	EKF	6	Ob	Video	PPP	
Zha	2	?	NP	NN	6	RVS	CCD	NLE	

* These projects use similar tracking methods as the Argo Project.
** depending on mode of operation; may or may not use triangulation

Table A.2: Commercial Video Based Motion Capture Systems

	Product Name	Supplier/Manufacturer	Country of Origin	Primary Uses	Contact
1	Advanced Space Vision System	Neptec Design Group Ltd.	Canada	Space robotics	Neptec, Kanata, Ontario (613) 599-7602
2	Eagle Eye	Kinetic Sciences Inc.	Canada	robotic navigation & calibration	www.kinetic.bc.ca
3	Ariel Performance Analysis System	Ariel Dynamics World-wide	USA	Human motion analysis Biomechanics	www.arielnet.com
4	Dynascope	Sirehna	France	model ship motions & robotics	sirehna@sirehna.ec-nantes.fr
5	Peak Motus	Peak Performance Technologies Inc.	USA	biomechanics	www.peakperform.com
6	Expert Vision	Motion Analysis Corp.	USA	biomechanics, industrial inspection & model ship motion	www.MotionAnalysis.com
7†	MacReflex	Qualisys AB	Sweden & USA	biomechanics, industrial inspection, robot calibration & model ship motions	www.qualisys.com
8	ProFlex or MultiTrax Pro	Qualisys AB &Adap-tive Optics of America	Sweden & USA	biomechanics, industrial inspection & robot calibration	www.qualisys.com www.aoainc.com
9	Vicon 370	Oxford Metrics Ltd.	UK & USA	biomechanics & model ship motions	www.metrics.co.uk
10	Lavic	Emtec Co. Ltd.	Japan	biomechanics, industrial inspection, robot calibration & model ship motions	Emtec Co. Ltd., Tokyo, Japan 03-3449-3 7 2 1
11	Videomex	Columbus Instruments International	USA	biology studies	www.colinst.com
12	Optofollow	Ya-Man Ltd.	Japan	industrial inspection	Ya-Man, San Jose, Calif (408) 559-9100
13	Optotrak	Northern Digial Inc.	Canada	biomechanics & robot calibration	www.ndigital.com
14	Selspot 3D Motion Analysis	Selspot AB	Sweden & USA	biomechanics, industrial inspection, robot calibration & model ship motions	Innovision Systems, Warren, MI (810) 751-0600

† the commercial system used to test the Argo Project

Table A.3: Commercial Video Based Motion Capture Systems: Specifications

	Product Name	Number of Cameras	Type of Camera	Sample Rate	Lighting	Type of Marker	Number of Markers	Host Platform	Accuracy and Resolution
1	Advanced Space Vision System	1 - 4	CCD	NP	NOL	HCBW EC&CT	NP	PC	NP
2	Eagle Eye	1	CCD	2-10 fps	NOL	HCBW	32x8	Mac	1:6000 X; 1:4000 Y; 2% Z
3	Ariel Performance Analysis System	2-9	CCD	60 fps	NOL	FP + HILP		PC + FG	
4	Dynascope	2	CCD	25 fps	NOL	HCBW	8	PC	NP
5	Peak Motus	2-6	CCD	50-60 & 200	NOL+IRS	FP+RRS	NP	PC	NP
6	Expert Vision	1 - 8	CCD+IR	60 & 240 fps	NOL+IRS	RRS	100	SGI or Sun + PP	1:2000
7†	MacReflex	1-7	CCD+IR	50 & 60	NOL+IRS	RRS	50	Mac + PP	1:30000
8	ProFlex or MultiTrax Pro	2-7	CCD+IR	240 & 1000	NOL+IRS	RRS	200	PC + PP	1:37000 @ 240 1:12333 @ 1000
9	Vicon 370	4-7	CCD+IR	60 fps	NOL+IRS	RRS	50+	PC + PP	NP
10	Lavic	2	CCD-clr	60 fps	NOL	HCCM	8	PC + PP	NP
11	Videomex	1	CCD-clr	60 fps	NOL	HCCM	6	PC	NP
12	Optofollow	1-4	IDT	3 kHz	DC light-ing	CAIR	100	PP	99.05%FOV; 0.001%FOV
13	Optotrak	1-8	3LCCD	3.5 kHz	NOL	CAIR	256	PC	1:22500 (X,Y); 1:225000
14	Selspot 3D Motion Analysis	1 - 16	PSD	10 kHz	NOL	CAIR	120	PC	99.5% FOV; 0.025%FOV

† the commercial system used to test the Argo Project

Table A.4: Commercial Video Based Motion Capture Systems: Tracking and Data Reduction

	Product Name	Camera Model	Lens Distortion Correction	Tracking Method	Analogue Input Channels	Special Options	Real-time Tracking	Point Tracking	Rigid Body tracking	Triangulation
1	Advanced Space Vision System	NP	NP	NP ref Pinkney & Perratt NRC-Ottawa	No		Yes	No	Yes	No
2	Eagle Eye	PPP	Yes	model based tracking	No		No	No	Yes	No
3	Ariel Performance Analysis System	DLT	Yes	predict next point	16 x 12bit @20 kHz 5 kHz/ch. max	Pan module available	No	Yes	No	Yes
4	Dynascope	NP	NP	NP	No		NP	Yes	No	Yes
5	Peak Motus	DLT	Yes	Dainis	8 + 32	Pan & Tilt of cameras	Yes	Yes	No	Yes
6	Expert Vision	NP	Yes	3-D ray tracing	32		Yes	Yes	No	Yes
7†	MacReflex	NP	Yes	NP	16 x 16 bit @500 Hz 16 digital inputs	2 analogue outputs	Yes	Yes	No & Yes*	Yes
8	ProFlex or MultiTrax Pro	NP	Yes	NP	16 x 16 bit @500 Hz 16 digital inputs	2 analogue outputs	Yes	Yes	No	Yes
9	Vicon 370	NP	NP	NP	64 x 12 bit @25kHz		NP	Yes	No	Yes
10	Lavic	NP	Yes	different colours	No	external trigger	Yes	Yes	No	Yes
11	Videomex	NP	NP	different colours	No	external trigger	NP	Yes	No	No
12	Optofollow	NP	Yes	Single active marker per sample	No		Yes	Yes	No	Yes
13	Optotrak	NP	NP	Single active marker per sample	No		Yes	Yes	Yes	Yes
14	Selspot 3D Motion Analysis	NP	NP	Single active marker per sample	16 - 48		Yes	Yes	No	Yes

† the commercial system used to test the Argo Project

*Special configuration for robot calibration

Appendix B: Homogeneous Transformations

In the study of ship motion, the transformation of coordinate frames from the world's to the ship's uses a roll, pitch, yaw rotation sequence; where roll is the angle (ψ_S^W) about the x axis of the world frame, pitch is the angle (θ_S^W) about the y axis of the world frame and yaw (ϕ_S^W) is the rotation about the z axis of the world frame. The world frame is a right handed coordinate system for the z axis up is positive.

The transformations presented in this appendix were derived using the symbolic math processor, Maple V R4.

B.1 Basic affine transformations

Rotation about x axis (roll)

$$rot_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi & 0 \\ 0 & \sin\psi & \cos\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \text{(B.1)}$$

Rotation about y axis (pitch)

$$rot_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \text{(B.2)}$$

Rotation about z axis (yaw)

$$rot_z = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \text{(B.3)}$$

Translation

$$trans = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \text{(B.4)}$$

B.2 Roll Pitch Yaw Translation Transformation

This transformation is typically used to describe the pose of a boat or an airplane in the world frame.

Transformation for a point in frame B into frame A.

$$RPYT = trans \cdot rot_z \cdot rot_y \cdot rot_x \ , \tag{B.5}$$

$$ {}_A RPYT_B = \begin{bmatrix} c\phi \cdot c\theta & -s\phi \cdot c\psi + c\phi \cdot s\theta \cdot s\psi & s\phi \cdot s\psi + c\phi \cdot s\theta \cdot c\psi & x_B^A \\ s\phi \cdot c\theta & c\phi \cdot c\psi + s\phi \cdot s\theta \cdot s\psi & -c\phi \cdot s\psi + s\phi \cdot s\theta \cdot c\psi & y_B^A \\ -s\theta & c\theta \cdot s\psi & c\theta \cdot c\psi & z_B^A \\ 0 & 0 & 0 & 1 \end{bmatrix} , \tag{B.6}$$

where:

$$c\psi = \cos(\psi_B^A) \ , \qquad c\theta = \cos(\theta_B^A) \ , \qquad c\phi = \cos(\phi_B^A) \ ,$$

$$s\psi = \sin(\psi_B^A) \ , \qquad s\theta = \sin(\theta_B^A) \ , \qquad s\phi = \sin(\phi_B^A) \ .$$

This transformation can be decomposed to its basic terms: [Fu 87]

Orientation (rotation) terms[1]:

$$\phi_B^A = \text{atan}\frac{RPYT_{1,1}}{RPYT_{2,1}} \ , \tag{B.7.a}$$

$$\psi_B^A = \text{atan}\frac{RPYT_{3,2}}{RPYT_{3,3}} \ , \tag{B.7.b}$$

$$TEMP = [rot_z]^{-1}[RPYT] \ , \tag{B.7.c}$$

$$\theta_B^A = \text{atan}\frac{-TEMP_{3,1}}{TEMP_{1,1}} \ . \tag{B.7.d}$$

Position (translation) terms:

$$x_B^A = RPYT_{1,4} \ , \tag{B.7.e}$$

$$y_B^A = RPYT_{2,4} \ , \tag{B.7.f}$$

$$z_B^A = RPYT_{3,4} \ , \tag{B.7.g}$$

---

1. For implementation use the MATLAB™ atan2 function for more reliable results.

B.3 Yaw Pitch Roll Translate Transformation

This transformation is typically used to describe the pose of a robot end effector in the world frame.

Transformation for a point in frame B into frame A.

$$YPRT = trans \cdot rot_x \cdot rot_y \cdot rot_z \ , \tag{B.8}$$

$$YPRT_B^A = \begin{bmatrix} c\phi c\theta & -s\phi c\theta & s\theta & x_B^A \\ c\phi s\theta s\psi + s\phi c\psi & -s\phi s\theta s\psi + c\phi c\psi & -c\phi s\psi & y_B^A \\ -c\phi s\theta c\psi + s\phi s\psi & s\phi s\theta c\psi + c\phi s\psi & c\phi c\psi & z_B^A \\ 0 & 0 & 0 & 1 \end{bmatrix} , \tag{B.9}$$

where:

$c\psi = \cos(\psi_B^A)$, $c\theta = \cos(\theta_B^A)$, $c\phi = \cos(\phi_B^A)$,

$s\psi = \sin(\psi_B^A)$, $s\theta = \sin(\theta_B^A)$, $s\phi = \sin(\phi_B^A)$.

This transformation can be decomposed to its basic terms: [Fu 87]

Orientation (rotation) terms:

$$\phi_B^A = \text{atan}\frac{-YPRT_{1,2}}{YPRT_{1,1}} \ , \tag{B.10.a}$$

$$\psi_B^A = \text{atan}\frac{YPRT_{2,3}}{YPRT_{3,3}} \ , \tag{B.10.b}$$

$$TEMP = [YPRT][rot_z]^{-1} \ , \tag{B.10.c}$$

$$\theta_B^A = \text{atan}\frac{TEMP_{1,3}}{TEMP_{1,1}} \ , \tag{B.10.d}$$

Position (translation) terms:

$$x_B^A = YPRT_{1,4} \ , \tag{B.10.e}$$

$$y_B^A = YPRT_{2,4} \ , \tag{B.10.f}$$

$$z_B^A = YPRT_{3,4} \ , \tag{B.10.g}$$

B.4 Pose extraction from triangulation results

Pose determination of one coordinate system with respect to another can be determined if positions of several points are knonw in both coordinate systems. A direct method can be used where the positions of feature points in the first frame are placed in a matrix, Equation (B.11), and the positions of the same feature points in the second frame are also place in a matrix, Equation (B.12).

$$[AMP] = \begin{bmatrix} x_1^A & x_2^A & \cdots & x_n^A & \cdots & x_N^A \\ y_1^A & y_2^A & \cdots & y_n^A & \cdots & y_N^A \\ z_1^A & z_2^A & \cdots & z_n^A & \cdots & z_N^A \\ 1 & 1 & \cdots & 1 & \cdots & 1 \end{bmatrix} , \tag{B.11}$$

$$[BMP] = \begin{bmatrix} x_1^B & x_2^B & \cdots & x_n^B & \cdots & x_N^B \\ y_1^b & y_2^B & \cdots & y_n^B & \cdots & y_N^B \\ z_1^B & z_2^B & \cdots & z_n^B & \cdots & z_N^B \\ 1 & 1 & \cdots & 1 & \cdots & 1 \end{bmatrix} , \tag{B.12}$$

It is important to ensure that the order that the feature points are placed is the same in both matrices. The general model for transforming several point in one frame of refernce to another is given in Equation (B.13).

$$[AMP] = {}_AH_B \cdot [BMP] , \tag{B.13}$$

In this case, the transformation, ${}_AH_B$, is the unknown. To solve for the unknown the postion matrix for the original frame is inverted and multiplied with the postion matrix of the destination frame. To determine orientation a minimum of four non-coplanar points are used. If more then four points are used, the postion matricies will not be rectangular. In this case the psuedo-inverse is used [Golub 89].

$$[{}_AH_B] = [AMP] \cdot [[[BMP]^T[BMP]]^{-1}[BMP]^T] , \tag{B.14}$$

The resultant [4 x 4] transformation matrix, ${}_AH_B$ in Equation (B.14), can then be decomposed into

its basic terms using the desired transformation model.

The least squares method uses the nonlinear equations for each feature point transformation as a constraint equation and solves for the six basic variables. Each feature point has three constraint equations, of the form of Equations (B.15), (B.16) and (B.17), specific to the transformation model being solved for. The roll, pitch, yaw, translation transformation produces the following constraint equations for a given point *n*.

$$x_n^A = c\phi \cdot c\theta \cdot x_n^B + (-s\phi \cdot c\psi + c\phi \cdot s\theta \cdot s\psi) \cdot y_n^B + (s\phi \cdot s\psi + c\phi \cdot s\theta \cdot c\psi) \cdot z_n^B + x_B^A \ , \quad \text{(B.15)}$$

$$y_n^A = s\phi \cdot c\theta \cdot x_n^B + (c\phi \cdot c\psi + s\phi \cdot s\theta \cdot s\psi) \cdot y_n^B + (-c\phi \cdot s\psi + s\phi \cdot s\theta \cdot c\psi) \cdot z_n^B + y_B^A \ , \quad \text{(B.16)}$$

$$z_n^A = -s\theta \cdot x_n^B + c\theta \cdot s\psi \cdot y_n^B + c\theta \cdot c\psi \cdot z_n^B + z_B^A \ , \quad \text{(B.17)}$$

where:

$$c\psi = \cos(\psi_B^A) \ , \qquad c\theta = \cos(\theta_B^A) \ , \qquad c\phi = \cos(\phi_B^A) \ ,$$

$$s\psi = \sin(\psi_B^A) \ , \qquad s\theta = \sin(\theta_B^A) \ , \qquad s\phi = \sin(\phi_B^A) \ .$$

Appendix C: Direct Linear Transformation (DLT)

The Direct Linear Transformation (DLT) was developed as a linearized camera model for use in photogrammetry for non-metric off the shelf cameras, in the late 1960's and early 1970's. The DLT was developed to triangulate the positions of objects from image coordinates extracted from stereo images with a stereo-photo-comparator or analytical plotter [Karara 89 & Miller 80].

C.1 Direct Linear Transformation Camera Model

The derivation of the Direct Linear Transformation (DLT) is based on the pin hole camera model and the colinear theory which is shown in Figure C.1 and given Equation (C.1), repeated here from chapter 3 for convenience of the reader. A detailed presentation of the DLT can be found in [Aziz 74, Karara 89, Marzan 76 & Shapiro 77].

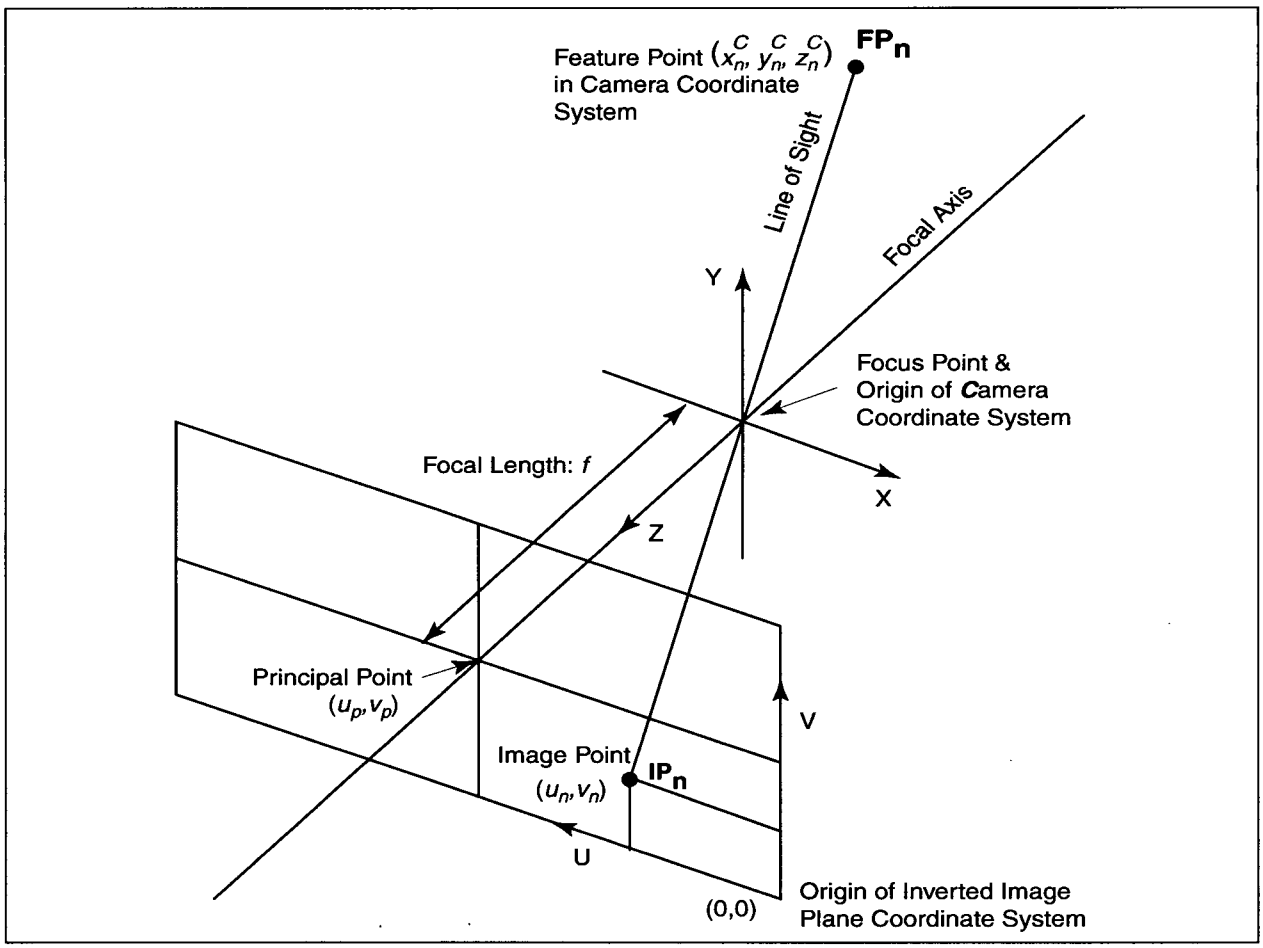


Figure C.1: Pin Hole Camera

Equation (C.1) [Gosine 96 & Weng 92a] describes a plane perspective projection in a viewer centred coordinate system.

$$u_n + du_n - u_p = \frac{x_n^C \cdot f_u}{z_n^C} \quad , \tag{C.1.a}$$

$$v_n + dv_n - v_p = \frac{y_n^C \cdot f_v}{z_n^C} \quad , \tag{C.1.b}$$

where:

- (u_n, v_n) = image coordinates of feature point n in camera C,
- (u_p, v_p) = image principal point in camera C,
- (f_u, f_v) = focal lengths in horizontal and vertical planes of camera C,
- (du_n, dv_n) = systematic error terms corresponding to image coordinates n in camera C,
- (x_n^C, y_n^C, z_n^C) = coordinates of feature point n in camera coordinate frame C.

(Note: All the above terms are in the camera frame.)

The model, presented in Figure C.1 and Equation (C.1), uses a viewer centred coordinate system. If the world coordinate system and the camera system do not coincide, a transformation is required to move feature points (FP_n) from the world frame to the camera frame, whose origin is the focal point, so that they can be projected onto the image plane This transformation is vector addition, followed by a rotation to re-orient the new vector from the world frame to the orientation for the camera frame. This transformation is represented in Figure C.2 and Equation (C.2). The rotation transformation in Equation (C.2), is a rotation about the z axis of the world frame, followed by a rotation about the y axis of the world frame, and then a rotation about the x axis of the world frame. Details of its derivation can be found in Appendix B. From this point this rotation transformation will be represented as the matrix R.

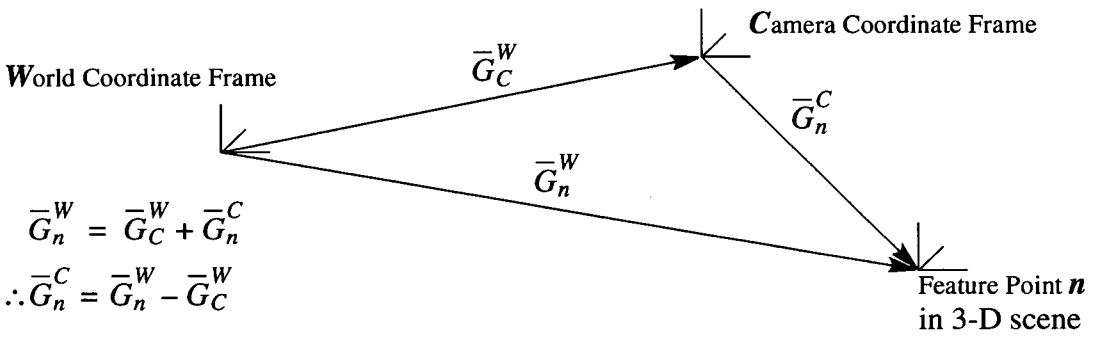
$$\begin{bmatrix} x_n^C \\ y_n^C \\ z_n^C \end{bmatrix} = \begin{bmatrix} c\phi c\theta & -s\phi c\theta & s\theta \\ c\phi s\theta s\psi + s\phi c\psi & -s\phi s\theta s\psi + c\phi c\psi & -c\phi s\psi \\ -c\phi s\theta c\psi + s\phi s\psi & s\phi s\theta c\psi + c\phi s\psi & c\theta c\psi \end{bmatrix} \cdot \begin{bmatrix} x_n^W - x_C^W \\ y_n^W - y_C^W \\ z_n^W - z_C^W \end{bmatrix} , \quad \text{(C.2)}$$

where:

$c\psi = \cos(\psi_C^W)$ $\quad c\theta = \cos(\theta_C^W)$ $\quad c\phi = \cos(\phi_C^W)$

$s\psi = \sin(\psi_C^W)$ $\quad s\theta = \sin(\theta_C^W)$ $\quad s\phi = \sin(\phi_C^W)$

($(\psi_C^W, \theta_C^W, \phi_C^W)$ = orientation of camera frame C in the *W*orld coordinate frame.)



where:

$\bar{G}_n^C = (x_n^C, y_n^C, z_n^C)$ = coordinates of feature point *n* in camera *C*

$\bar{G}_n^W = (x_n^W, y_n^W, z_n^W)$ = coordinates of feature point *n* in the *W*orld coordinate frame.

$\bar{G}_C^W = (x_C^W, y_C^W, z_C^W)$ = position of focal point (camera frame C) in the *W*orld coordinate frame.

Figure C.2: Feature point position in the *C*amera's and the *W*orld coordinates frames

Combining Equations (C.1) and (C.2) results in a camera model that includes both the intrinsic parameters from Equation (C.1) and the extrinsic parameters, camera pose, from Equation (C.2). The results of this combination can be found in Equation (C.3) [Aziz 74].

$$u_n + du_n = u_p - f_u \cdot \left(\frac{R_{1,1}(x_n^W - x_C^W) + R_{1,2}(y_n^W - y_C^W) + R_{1,3}(z_n^W - z_C^W)}{R_{3,1}(x_n^W - x_C^W) + R_{3,2}(y_n^W - y_C^W) + R_{3,3}(z_n^W - z_C^W)} \right) , \quad \text{(C.3.a)}$$

$$v_n + dv_n = v_p - f_v \cdot \left(\frac{R_{2,1}(x_n^W - x_C^W) + R_{2,2}(y_n^W - y_C^W) + R_{2,3}(z_n^W - z_C^W)}{R_{3,1}(x_n^W - x_C^W) + R_{3,2}(y_n^W - y_C^W) + R_{3,3}(z_n^W - z_C^W)} \right) , \quad \text{(C.3.b)}$$

Through algebraic manipulation Equation (C.3) can be represented as:

$$u_n + du_n = \frac{L_1 \cdot x_n^W + L_2 \cdot y_n^W + L_3 \cdot z_n^W + L_4}{L_9 \cdot x_n^W + L_{10} \cdot y_n^W + L_{11} \cdot z_n^W + 1} \quad , \tag{C.4.a}$$

$$v_n + dv_n = \frac{L_5 \cdot x_n^W + L_6 \cdot y_n^W + L_7 \cdot z_n^W + L_8}{L_9 \cdot x_n^W + L_{10} \cdot y_n^W + L_{11} \cdot z_n^W + 1} \quad , \tag{C.4.b}$$

where:

- $L_{1\text{-}11}$ = DLT parameters.

The expanded form of the individual DLT parameters are as follows in the Equation (C.5):

Let $\alpha = -(R_{3,1}x_C^W + R_{3,2}y_C^W + R_{3,3}z_C^W)$,

$$L_1 = (u_p R_{3,1} - f_u R_{1,1})/\alpha \quad , \tag{C.5.a}$$

$$L_2 = (u_p R_{3,2} - f_u R_{1,2})/\alpha \quad , \tag{C.5.b}$$

$$L_3 = (u_p R_{3,3} - f_u R_{1,3})/\alpha \quad , \tag{C.5.c}$$

$$L_4 = u_p + f_u(R_{1,1}x_C^W + R_{1,2}y_C^W + R_{1,3}z_C^W)/\alpha \quad , \tag{C.5.d}$$

$$L_5 = (v_p \cdot R_{1,1} - f_v \cdot R_{2,1})/\alpha \quad , \tag{C.5.e}$$

$$L_6 = (v_p R_{2,1} - f_v R_{2,2})/\alpha \quad , \tag{C.5.f}$$

$$L_7 = (v_p R_{3,1} - f_v R_{2,3})/\alpha \quad , \tag{C.5.g}$$

$$L_8 = v_p + f_v(R_{2,1}x_C^W + R_{2,2}y_C^W + R_{2,3}z_C^W)/\alpha \quad , \tag{C.5.h}$$

$$L_9 = R_{3,1}/\alpha \quad , \tag{C.5.i}$$

$$L_{10} = R_{3,2}/\alpha \quad , \tag{C.5.j}$$

$$L_{11} = R_{3,3}/\alpha \quad , \tag{C.5.k}$$

The DLT parameters in Equation (C.5) [Aziz 74] can be decomposed to obtain the camera extrinsic and intrinsic calibration parameters as expressed in Equation (C.6) [Aziz 74].

$$\text{Let } \beta = -1/\sqrt{(L_9^2 + L_{10}^2 + L_{11}^2)} \quad ,$$

$$u_p = (L_1L_9 + L_2L_{10} + L_3L_{11})\beta^2 \quad , \tag{C.6.a}$$

$$v_p = (L_5L_9 + L_6L_{10} + L_7L_{11})\beta^2 \quad , \tag{C.6.b}$$

$$f_u = \sqrt{(L_1^2 + L_2^2 + L_3^2)\beta^2 - u_p^2} \quad , \tag{C.6.c}$$

$$f_v = \sqrt{(L_5^2 + L_6^2 + L_7^2)\beta^2 - v_p^2} \quad , \tag{C.6.d}$$

$$\theta_C^W = \text{asin}(L_9\beta) \quad , \tag{C.6.e}$$

$$\psi_C^W = \text{atan}(-L_{10}/L_{11}) \quad , \tag{C.6.f}$$

$$R_{1,1} = \beta(u_pL_9 - L_1)/f_u \quad , \tag{C.6.g}$$

$$\phi_C^W = \text{acos}(R_{1,1})/\cos\theta_C^W \quad , \tag{C.6.h}$$

$$\begin{bmatrix} x_C^W \\ y_C^W \\ z_C^W \end{bmatrix} = \begin{bmatrix} L_1 & L_2 & L_3 \\ L_5 & L_6 & L_7 \\ L_9 & L_{10} & L_{11} \end{bmatrix}^{-1} \begin{bmatrix} L_4 \\ L_8 \\ 1 \end{bmatrix} \quad , \tag{C.6.i}$$

C.2 DLT Calibration Equations

The DLT camera model, described by Equations (C.4), projects the positions of a feature point in the world coordinate system onto the image plane of the camera. The eleven DLT parameters for a given camera m, L_{1-11}^m, can be determined with either a direct least squares method or with an iterative least squares method. The direct method assumes that the distortion correction terms are small and are therefore ignored. The iterative least squares method solves for the linearized DLT parameters as well as the non-linear distortion correction terms if they are included in the camera model. The iterative least squares method is more rigorous and will yield a better calibration at the cost of increased computation. Using the direct method to supply the initial guess for

the least squares problem should reduce the number of iterations required to obtain a solution.

The calibration procedure is to place a minimum of six markers at known positions in the world coordinate system in the field of view of the camera being calibrated. For calibrating cameras covering a large test scene, where a given camera can not view the entire scene, different markers can be used for each camera as long as the same world coordinate system is used to describe the position of the markers. The image coordinates corresponding to the centroids of the projections of the markers onto the image plane are then recorded. Knowing both the marker positions and the corresponding image coordiantes leaves only the eleven unknown DLT parameters in the constraint Equation (C.4).

The direct method of solving for the DLT parameters, L^m_{1-11}, puts the set of constraint equations into matrix form and then isolates the unknown paramters. The matrix form of the constraint equations is shown in Equation (C.7).

$$\begin{bmatrix} x_1^W & y_1^W & z_1^W & 1 & 0 & 0 & 0 & 0 & -u_1^m x_1^W & -u_1^m y_1^W & -u_1^m z_1^W \\ 0 & 0 & 0 & 0 & x_1^W & y_1^W & z_1^W & 1 & -v_1^m x_1^W & -v_1^m y_1^W & -v_1^m z_1^W \\ x_2^W & y_2^W & z_2^W & 1 & 0 & 0 & 0 & 0 & -u_2^m x_2^W & -u_2^m y_2^W & -u_2^m z_2^W \\ 0 & 0 & 0 & 0 & x_2^W & y_2^W & z_2^W & 1 & -v_2^m x_2^W & -v_2^m y_2^W & -v_2^m z_2^W \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_n^W & y_n^W & z_n^W & 1 & 0 & 0 & 0 & 0 & -u_n^m x_n^W & -u_n^m y_n^W & -u_n^m z_n^W \\ 0 & 0 & 0 & 0 & x_n^W & y_n^W & z_n^W & 1 & -v_n^m x_n^W & -v_n^m y_n^W & -v_n^m z_n^W \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_N^W & y_N^W & z_N^W & 1 & 0 & 0 & 0 & 0 & -u_N^m x_N^W & -u_N^m y_N^W & -u_N^m z_N^W \\ 0 & 0 & 0 & 0 & x_N^W & y_N^W & z_N^W & 1 & -v_N^m x_N^W & -v_N^m y_N^W & -v_N^m z_N^W \end{bmatrix} \cdot \begin{bmatrix} L_1^m \\ L_2^m \\ L_3^m \\ L_4^m \\ L_5^m \\ L_6^m \\ L_7^m \\ L_8^m \\ L_9^m \\ L_{10}^m \\ L_{11}^m \end{bmatrix} = \begin{bmatrix} u_1^m \\ v_1^m \\ u_2^m \\ v_2^m \\ \cdots \\ \cdots \\ u_n^m \\ v_n^m \\ \cdots \\ \cdots \\ u_N^m \\ v_N^m \end{bmatrix}, \qquad \text{(C.7)}$$

The general form of the above calibration equation is: $[MP][L] = [IC]$

where:

- $[MP]$ = marker parameters matrix (2N by 11)
- $[L]$ = DLT parameters vector (11 by 1)
- $[IC]$ = image coordinates vector (2N by 1)

Premultiplying both sides of this equations by the inverse of the marker paramter matrix, $[MP]$ isolates the unknown parameters on one side of the equation.

$$[MP]^{-1}[MP][L] = [MP]^{-1}[IC] = [L] \quad , \tag{C.8}$$

Due to the rectangular nature of the marker parameter matrix, $[MP]$, a psuedo-inverse is necessary to solve Equation (C.8). The psuedo-inverse is an overdetermined least-squares solution [Golub 89]. Including the psuedo-inverse into Equation (C.8) the form of the solution becomes Equation (C.9).

$$[L] = ([MP]^{T}[MP])^{-1}[MP]^{T}[IC] \quad , \tag{C.9}$$

C.3 Triangulation using DLT

The advantage of the DLT is that triangulation of the position in the world coordinate system of any marker is reduced to a simple matrix operation. To triangulate a marker's location a minimum, of two points of view are needed. Additional data from other images can be included to minimize the error in the triangulation.

$$\begin{bmatrix} (L_1^1 - L_9^1 u_n^1) & (L_2^1 - L_{10}^1 u_n^1) & (L_3^1 - L_{11}^1 u_n^1) \\ (L_5^1 - L_9^1 v_n^1) & (L_6^1 - L_{10}^1 v_n^1) & (L_7^1 - L_{11}^1 v_n^1) \\ (L_1^2 - L_9^2 u_n^2) & (L_2^2 - L_{10}^2 u_n^2) & (L_3^2 - L_{11}^2 u_n^2) \\ (L_5^2 - L_9^2 v_n^2) & (L_6^2 - L_{10}^2 v_n^2) & (L_7^2 - L_{11}^2 v_n^2) \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ (L_1^m - L_9^m u_n^m) & (L_2^m - L_{10}^m u_n^m) & (L_3^m - L_{11}^m u_n^m) \\ (L_5^m - L_9^m v_n^m) & (L_6^m - L_{10}^m v_n^m) & (L_7^m - L_{11}^m v_n^m) \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ (L_1^M - L_9^M u_n^M) & (L_2^M - L_{10}^M u_n^M) & (L_3^M - L_{11}^M u_n^M) \\ (L_5^M - L_9^M v_n^M) & (L_6^M - L_{10}^M v_n^M) & (L_7^M - L_{11}^M v_n^M) \end{bmatrix} \begin{bmatrix} x_n^W \\ y_n^W \\ z_n^W \end{bmatrix} = \begin{bmatrix} (u_n^1 - L_4^1) \\ (v_n^1 - L_8^1) \\ (u_n^2 - L_4^2) \\ (v_n^2 - L_8^2) \\ \cdots \\ \cdots \\ (u_n^m - L_4^m) \\ (v_n^m - L_8^m) \\ \cdots \\ \cdots \\ (u_n^M - L_4^M) \\ (v_n^M - L_8^M) \end{bmatrix}, \qquad \text{(C.10)}$$

The general form of the above calibration equation is: $[AB][Pt] = [BC]$
where:

- $[AB]$ = triangulation matrix (2M by 3)
- $[Pt]$ = point position vector (3 by 1)
- $[BC]$ = triangulation vector (2M by 1)

It is assumed that the marker can be seen in all fields of view (FOV). In the event that the marker does not appear in an image but is still visible in at least two images, the rows in the triangulation matrix and vector corresponding to the image where it is not visible are removed. The position of marker n in the *W*orld coordinate frame is found by inverting the triangulation matrix and then post-multiplying with the triangulation vector, Equation (C.11).

$$[Pt] = [AB]^{-1}[BC] \quad , \tag{C.11}$$

The rectangular nature of the triangulation matrix, $[AB]$, requires a pseudo-inverse, Equation (C.12), to invert it.

$$[Pt] = [[[AB]^{T}[AB]]^{-1}[AB]^{T}][BC] \quad , \tag{C.12}$$

Appendix D: Initialization of the State Error Covariance for the EKF

This appendix presents the initialization of the state error covariance matrix, P_0, necessary to initiate the extended Kalman filter (EKF). The initial state error covariance matrix, P_0, depends on the dynamics of the system and the observations made of that system. In the case of the Argo Project the dynamic model assumes a constant acceleration linear relationship between states. With the Argo Project it is not possible to directly observe the state of the system. The observations made, measures the pose of the mobile rigid body (MRB) and can not measure any of the higher order terms of the pose.

Since the dynamics model is the same for each degree of freedom, the derivation of the intial state error covariance matrix, P_0, is based on a single degree of freedom and repeated for each one.

An effort is made to make the initial state error covariance matrix, P_0, as accurate as possible but as the EKF adjusts the state error covariance matrix, P_k, it is not necessary that the initial value perfectly match the system, as it is only a starting point.

D.1 Nomenclature

$NDOF$	number of degrees of freedom (default = 6)
$NORD$	numerical order of model (default = 3)
M	number of cameras
N	number of markers on target frame
q	number of state variables = $NORD \cdot NDOF = 18$
r	number of estimated observations = $2 \cdot M \cdot N$
k	discrete time index ($k \in$ integer set)
Γ	error covariance matrix for a single DOF $[NORD \times NORD]$
P_k	estimate (error) covariance matrix $[q \times q]$
s_k	state vector $[q \times 1]$
$s_k^{\dagger}$	estimated state vector $[q \times 1]$
ξ_k	observation noise vector $[r \times 1]$
ω_3	system noise at interval 3 for a given DOF

τ sample period

$E\langle\ \rangle$ expected value operator

D.1.1 Initial State Error Covariance Model

Observations of the system only provides position information and all higher order state variables, such as velocity and acceleration, are inferred from the position data. The observed data is assumed to contain some white noise, represented by ξ_k.

$$x_3 = x_3 + \xi_3 \quad , \tag{D.1}$$

The inferred velocity term is:

$$\dot{x}_3 = \frac{(x_3 + \xi_3) - (x_2 + \xi_2)}{\tau} \quad , \tag{D.2}$$

The corresponding acceleration term is:

$$\ddot{x}_3 = \frac{\dfrac{(x_3 + \xi_3) - (x_2 + \xi_2)}{\tau} - \dfrac{(x_2 + \xi_2) - (x_1 + \xi_1)}{\tau}}{\tau} \quad , \tag{D.3}$$

By collecting like terms Equation (D.3) can be reduced to:

$$\ddot{x}_3 = \frac{(x_3 + \xi_3) - 2(x_2 + \xi_2) - (x_1 + \xi_1)}{\tau^2} \quad , \tag{D.4}$$

The EKF generates an estimate of the position term that corresponds to the above observation terms:

$$\hat{x}_3 = x_3 \quad , \tag{D.5}$$

The estimate of the velocity term is:

$$\dot{\hat{x}}_3 = \frac{x_3 - x_2}{\tau} \quad , \tag{D.6}$$

The dynamics model used in the Argo Project assumes that deficiencies in the estimation process are modelled as noise introduced in the estimate of the acceleration term is:

$$\ddot{\hat{x}}_3 = \frac{\dfrac{x_3 - x_2}{\tau} - \dfrac{x_2 - x_1}{\tau}}{\tau} + \omega_3 = \frac{x_3 - 2x_2 - x_1}{\tau^2} + \omega_3 \quad , \tag{D.7}$$

The general form of a third order state vector for a single degree of freedom system is:

$$s = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} , \tag{D.8}$$

Therefore, the observed state of the system from Equations (D.1) through (D.4), and Equation (D.8) is:

$$s_3 = \begin{bmatrix} x_3 + \xi_3 \\ \dfrac{(x_3 + \xi_3) - (x_2 + \xi_2)}{\tau} \\ \dfrac{(x_3 + \xi_3) - 2(x_2 + \xi_2) + (x_1 + \xi_1)}{\tau^2} \end{bmatrix} , \tag{D.9}$$

The estimated state vector for a single degree of freedom is the collection of Equations (D.5), (D.6) and (D.7), is:

$$\hat{s}_3 = \begin{bmatrix} x_3 \\ \dfrac{x_3 - x_2}{\tau} \\ \dfrac{x_3 - 2x_2 + x_1}{\tau^2} + \omega_3 \end{bmatrix} , \tag{D.10}$$

Taking the difference between the observed and estimated states of the system, yeilds a vector of state error terms:

$$s_3 - \hat{s}_3 = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad , \tag{D.11}$$

where:

$$a = (x_3 + \xi_3) - x_3 = \xi_3 \quad , \tag{D.12}$$

$$b = \frac{(x_3 + \xi_3) - (x_2 + \xi_2)}{\tau} - \frac{x_3 - x_2}{\tau} = \frac{\xi_3 - \xi_2}{\tau} \quad , \tag{D.13}$$

$$c = \frac{(x_3 + \xi_3) - 2(x_2 + \xi_2) + (x_1 + \xi_1)}{\tau^2} - \left(\frac{x_3 - 2x_2 + x_1}{\tau^2} + \omega_3 \right) = \frac{\xi_3 - 2\xi_2 + \xi_1}{\tau^2} - \omega_3 \quad , \tag{D.14}$$

As previously stated in Section 4.2.4, the definition of the state error covariance is:

$$P_k = E\langle (s_k - \hat{s}_k)(s_k - \hat{s}_k)^T \rangle \tag{D.15}$$

The result of combining Equations (D.11) through (D.14), and Equation (D.15) is:

$$P = \begin{bmatrix} E\langle a^2 \rangle & E\langle ab \rangle & E\langle ac \rangle \\ E\langle ab \rangle & E\langle b^2 \rangle & E\langle bc \rangle \\ E\langle ac \rangle & E\langle bc \rangle & E\langle c^2 \rangle \end{bmatrix} \tag{D.16}$$

where:

$$a^2 = \xi_3^2 \quad , \tag{D.17}$$

$$ab = \frac{\xi_3^2}{\tau} - \frac{\xi_2 \xi_3}{\tau} \quad , \tag{D.18}$$

$$ac = \frac{\xi_3^2 - 2\xi_2 \xi_3 + \xi_1 \xi_3}{\tau^2} - \omega_3 \xi_3 \quad , \tag{D.19}$$

$$bc = \frac{\xi_3^2}{\tau^3} - 3\frac{\xi_2\xi_3}{\tau^3} + \frac{\xi_1\xi_3}{\tau^3} - \frac{\omega_3\xi_3}{\tau} + 2\frac{\xi_2^2}{\tau^3} - \frac{\xi_1\xi_2}{\tau^3} + \frac{\omega_3\xi_2}{\tau} \quad , \tag{D.20}$$

$$b^2 = \frac{\xi_3^2}{\tau^2} - 2\frac{\xi_2\xi_3}{\tau^2} + \frac{\xi_2^2}{\tau^2} \quad , \tag{D.21}$$

$$c^2 = \frac{\xi_3^2}{\tau^4} - 4\frac{\xi_2\xi_3}{\tau^4} + 2\frac{\xi_1\xi_3}{\tau^4} - 2\frac{\omega_3\xi_3}{\tau^2} + 4\frac{\xi_2^2}{\tau^4} - 4\frac{\xi_1\xi_2}{\tau^4} + 2\frac{\omega_3\xi_2}{\tau^2} + \frac{\xi_1^2}{\tau^4} - \frac{\omega_3\xi_1}{\tau^2} + \omega_3^2 \quad , \tag{D.22}$$

Assuming that the above noise terms are uncorrelated, then the following constraints are applied for determining the expected values of the above terms.

$$E\langle \xi_j \xi_k^T \rangle = \begin{pmatrix} \sigma_\xi^2 & , j = k \\ 0 & , j \neq k \end{pmatrix} \qquad \text{for all } j \text{ and } k \quad , \tag{D.23}$$

$$E\langle \omega_j \omega_k^T \rangle = \begin{pmatrix} \sigma_\omega^2 & , j = k \\ 0 & , j \neq k \end{pmatrix} \qquad \text{for all } j \text{ and } k \quad , \tag{D.24}$$

$$E\langle \xi_j \omega_k^T \rangle = 0 \qquad \text{for all } j \text{ and } k \quad , \tag{D.25}$$

$$E\langle \xi_k \rangle = E\langle \omega_k \rangle = 0 \qquad \text{for all } k> \quad , \tag{D.26}$$

Applying the constraints defined in Equations (D.23) through (D.26), the expected values of the products in Equations (D.17) through (D.22) are:

$$E\langle a^2 \rangle = \sigma_{\xi_3}^2 \quad , \tag{D.27}$$

$$E\langle b^2 \rangle = \frac{\sigma_{\xi_2}^2}{\tau^2} + \frac{\sigma_{\xi_3}^2}{\tau^2} \quad , \tag{D.28}$$

$$E\langle ab \rangle = \frac{\sigma_{\xi_3}^2}{\tau} \quad , \tag{D.29}$$

$$E\langle ac \rangle = \frac{\sigma_{\xi_3}^2}{\tau^2} \quad , \tag{D.30}$$

$$E\langle bc\rangle = 2\frac{\sigma_{\xi_2}^2}{\tau^3} + \frac{\sigma_{\xi_3}^2}{\tau^3} \quad , \tag{D.31}$$

$$E\langle c^2\rangle = \frac{\sigma_{\xi_1}^2}{\tau^3} + 4\frac{\sigma_{\xi_2}^2}{\tau^3} + \frac{\sigma_{\xi_3}^2}{\tau^3} + \sigma_{\omega_3}^2 \quad , \tag{D.32}$$

Assuming noise levels remain constant the following constraints can be added, Equations (D.33) and (D.34):

$$\sigma_{\omega_3}^2 = \sigma_{\omega}^2 \quad , \tag{D.33}$$

$$\sigma_{\xi_1}^2 = \sigma_{\xi_2}^2 = \sigma_{\xi_3}^2 = \sigma_{\xi}^2 \quad , \tag{D.34}$$

Applying Equations (D.33) and (D.34) to Equations (D.27) through (D.32), the expected values of the "error products" can be further reduced to Equations (D.35) through (D.40).

$$E\langle a^2\rangle = \sigma_{\xi}^2 \quad , \tag{D.35}$$

$$E\langle b^2\rangle = 2\frac{\sigma_{\xi}^2}{\tau^2} \quad , \tag{D.36}$$

$$E\langle ab\rangle = \frac{\sigma_{\xi}^2}{\tau} \quad , \tag{D.37}$$

$$E\langle ac\rangle = \frac{\sigma_{\xi}^2}{\tau^2} \quad , \tag{D.38}$$

$$E\langle bc\rangle = 3\frac{\sigma_{\xi}^2}{\tau^3} \quad , \tag{D.39}$$

$$E\langle c^2\rangle = 6\frac{\sigma_{\xi}^2}{\tau^3} + \sigma_{\omega}^2 \quad , \tag{D.40}$$

Resulting in the initial estimate of the state error covariance for a third order, single degree of freedom, system:

$$\Gamma = \begin{bmatrix} \sigma_\xi^2 & \frac{\sigma_\xi^2}{\tau} & \frac{\sigma_\xi^2}{\tau^2} \\ \frac{\sigma_\xi^2}{t} & 2\frac{\sigma_\xi^2}{\tau^2} & 3\frac{\sigma_\xi^2}{\tau^3} \\ \frac{\sigma_\xi^2}{\tau^2} & 3\frac{\sigma_\xi^2}{\tau^3} & 6\frac{\sigma_\xi^2}{\tau^4} + \sigma_\omega^2 \end{bmatrix}, \tag{D.41}$$

For a third order, six degrees of freedom system P_k is an [18 x 18] square matrix. With the current system and observation covariance models, the general form of the intial state error covariance, repeats the single DOF model for each subsequent DOF. It should be noted that this model does not include cross-coupling terms.

$$P_0 = \begin{bmatrix} \Gamma & 0 & 0 & 0 & 0 & 0 \\ 0 & \Gamma & 0 & 0 & 0 & 0 \\ 0 & 0 & \Gamma & 0 & 0 & 0 \\ 0 & 0 & 0 & \Gamma & 0 & 0 \\ 0 & 0 & 0 & 0 & \Gamma & 0 \\ 0 & 0 & 0 & 0 & 0 & \Gamma \end{bmatrix}, \tag{D.42}$$

Appendix E: Fully expanded observation model equations

This appendix contains all the equations necessary to generate the estimated image coordinates vector and the cooresponding linearized observation model, implemented in the EKF tracking module. All of the equations presented herein were derived using a symbolic math processor software pacakage, Maple V R4. Some of the basic equations used herein are presented in Appendices B and C.

The estimate of the image coordinates of corresponding markers for each camera and the linearized observation matrix are generated using the estimate of the state vector from the dynamics model in the EKF and the known geometry of the target attached to the MRB. Since the camera model is calibrated to the world coordinate frame, it is necessary to transform the known marker positions in the MRB coordinate frame to the World coordinate frame as expressed in Equation (E.1) This transformation, a roll-pitch-yaw-translation, uses the estimated pose of the MRB in the World coordinate frame as its parameters.

$$\begin{bmatrix} x_n^W \\ y_n^W \\ z_n^W \\ 1 \end{bmatrix} = \begin{bmatrix} c\phi \cdot c\theta & -s\phi \cdot c\psi + c\phi \cdot s\theta \cdot s\psi & s\phi \cdot s\psi + c\phi \cdot s\theta \cdot c\psi & x_{MRB}^W \\ s\phi \cdot c\theta & c\phi \cdot c\psi + s\phi \cdot s\theta \cdot s\psi & -c\phi \cdot s\psi + s\phi \cdot s\theta \cdot c\psi & y_{MRB}^W \\ -s\theta & c\theta \cdot s\psi & c\theta \cdot c\psi & z_{MRB}^W \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_n^{MRB} \\ y_n^{MRB} \\ z_n^{MRB} \\ 1 \end{bmatrix}, \qquad \text{(E.1)}$$

where:

$$c\phi = \cos(\phi_{MRB}^W) \qquad c\psi = \cos(\psi_{MRB}^W) \qquad c\theta = \cos(\theta_{MRB}^W)$$

$$s\phi = \sin(\phi_{MRB}^W) \qquad s\psi = \sin(\psi_{MRB}^W) \qquad s\theta = \sin(\theta_{MRB}^W)$$

- $(x_n^{MRB}, y_n^{MRB}, z_n^{MRB})$ = position of marker *n* in th *MRB* coordinate frame.
- (y_n^W, y_n^W, z_n^W) = positon of marker *n* in the *W*orld coordinate frame.
- $(\psi_{MRB}^W, \theta_{MRB}^W, \phi_{MRB}^W, x_{MRB}^W, y_{MRB}^W, z_{MRB}^W)$ = pose of *MRB* coordinate frame in the *W*orld coordinate frame.

The marker locations in the world coordinate frame are transformed into image coordinates for each point of view using the DLT camera model expressed in Equation (E.2) [Aziz 74].

$$u_n^m = \frac{L_1^m \cdot x_n^W + L_2^m \cdot y_n^W + L_3^m \cdot z_n^W + L_4^m}{L_9^m \cdot x_n^W + L_{10}^m \cdot y_n^W + L_{11}^m \cdot z_n^W + 1} \quad , \tag{E.2.a}$$

$$v_n^m = \frac{L_5^m \cdot x_n^W + L_6^m \cdot y_n^W + L_7^m \cdot z_n^W + L_8^m}{L_9^m \cdot x_n^W + L_{10}^m \cdot y_n^W + L_{11}^m \cdot z_n^W + 1} \quad , \tag{E.2.b}$$

where:

- (u_n^m, v_n^m) = image coordinates of marker n in camera m.
- L_{1-11}^m = DLT parameters for camera m.
- (x_n^W, y_n^W, z_n^W) = position of marker, n in the world coordinate frame, defined in Equation (E.1).

Explicitly the terms that describe the position of the marker in the World frame from knowing the pose of the MRB with respect to the World frame are shown in the equation group E.3. This is merely an expansion of the terms from equation E.1.

$$\begin{aligned} x_n^W &= \cos(\phi_{MRB}^W)\cos(\theta_{MRB}^W)x_n^{MRB} \\ &+ (-\sin(\phi_{MRB}^W)\cos(\psi_{MRB}^W) + \cos(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\sin(\psi_{MRB}^W))y_n^{MRB} \\ &+ (\sin(\phi_{MRB}^W)\sin(\psi_{MRB}^W) + \cos(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\cos(\psi_{MRB}^W))z_n^{MRB} \\ &+ x_{MRB}^W \end{aligned} \quad , \tag{E.3.a}$$

$$\begin{aligned} y_n^W &= \sin(\phi_{MRB}^W)\cos(\theta_{MRB}^W)x_n^{MRB} \\ &+ (\cos(\phi_{MRB}^W)\cos(\psi_{MRB}^W) + \sin(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\sin(\psi_{MRB}^W))y_n^{MRB} \\ &+ (-\cos(\phi_{MRB}^W)\sin(\psi_{MRB}^W) + \sin(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\cos(\psi_{MRB}^W))z_n^{MRB} \\ &+ y_{MRB}^W \end{aligned} \quad , \tag{E.3.b}$$

$$\begin{aligned} z_n^W &= -\sin(\theta_{MRB}^W)x_n^{MRB} + \cos(\theta_{MRB}^W)\sin(\psi_{MRB}^W)y_n^{MRB} \\ &+ \cos(\theta_{MRB}^W)\cos(\psi_{MRB}^W)z_n^{MRB} + z_{MRB}^W \end{aligned} \quad , \tag{E.3.c}$$

Equations (E.1) and (E.2) are combined to estimate the image coordinates of a marker n in camera m using the estimated MRB pose.

The image coordinate terms $\left(u_n^{m}, v_n^{m}\right)$ form the observation estimation vector, . The observation estimation vector is assembled using loop structure:

for m = 1 to M

for n = 1 to N

$$\hat{g}_k^{\dagger}(2 \times N \times (m-1) + 2 \times n - 1) = u_n^m \quad , \tag{E.4.a}$$

$$\hat{g}_k^{\dagger}(2 \times N \times (m-1) + 2 \times n) = v_n^m \quad , \tag{E.4.b}$$

end

end

The linearized observation model is obtained by taking the first partial derivative of the observation estimation vector with respect to the state vector, Equation (E.5) [Grewal 93].

$$C_k \cong \left.\frac{\partial g_k}{\partial s}\right|_{s = \hat{s}_k^{\dagger}} = \left.\frac{\partial}{\partial s} c(s, k)\right|_{s = \hat{s}_k^{\dagger}} \quad , \tag{E.5}$$

Since the observation equations are based solely on pose terms and do not include any higher order terms the derivatives of the image coordinates with respect to these higher order terms are zero. The resulting matrix is made up of alternating rows, corresponding to the alternating u_n and v_n terms in , $\hat{g}_k^{\dagger}$. The rows are of the following form, Equations (E.6) and (E.7):

Row $2 \times N \times (m-1) + 2 \times n - 1$ of the linearized observation matrix, C_k.

$$\left[\frac{\partial u_n^m}{\partial \psi_{MRB}^W} \; 0 \; 0 \; \frac{\partial u_n^m}{\partial \theta_{MRB}^W} \; 0 \; 0 \; \frac{\partial u_n^m}{\partial \phi_{MRB}^W} \; 0 \; 0 \; \frac{\partial u_n^m}{\partial x_{MRB}^W} \; 0 \; 0 \; \frac{\partial u_n^m}{\partial y_{MRB}^W} \; 0 \; 0 \; \frac{\partial u_n^m}{\partial z_{MRB}^W} \; 0 \; 0\right] \quad , \tag{E.6}$$

Row $2 \times N \times (m-1) + 2 \times n$ of the linearized observation matrix.

$$\left[\frac{\partial v_n^m}{\partial \psi_{MRB}^W} \; 0 \; 0 \; \frac{\partial v_n^m}{\partial \theta_{MRB}^W} \; 0 \; 0 \; \frac{\partial v_n^m}{\partial \phi_{MRB}^W} \; 0 \; 0 \; \frac{\partial v_n^m}{\partial x_{MRB}^W} \; 0 \; 0 \; \frac{\partial v_n^m}{\partial y_{MRB}^W} \; 0 \; 0 \; \frac{\partial v_n^m}{\partial z_{MRB}^W} \; 0 \; 0\right] \quad , \tag{E.7}$$

The explicit forms of the partial derivative terms in these two vectors are expressed in the Equations (E.8) through (E.13).

$$\frac{\partial u_n^m}{\partial \psi_{MOB}^W} = \frac{(L_1^m T_8 + L_2^m T_9 + L_3^m T_{10})}{(L_9^m T_7 + L_{10}^m T_6 + L_{11}^m T_5 + 1)} - \frac{T_{11}(L_9^m T_8 + L_{10}^m T_9 + L_{11}^m T_{10})}{(L_9^m T_7 + L_{10}^m T_6 + L_{11}^m T_5 + 1)^2}, \tag{E.8.a}$$

$$\frac{\partial v_n^m}{\partial \psi_{MOB}^W} = \frac{(L_5^m T_8 + L_6^m T_9 + L_7^m T_{10})}{(L_9^m T_7 + L_{10}^m T_6 + L_{11}^m T_5 + 1)} - \frac{T_{12}(L_9^m T_8 + L_{10}^m T_9 + L_{11}^m T_{10})}{(L_9^m T_7 + L_{10}^m T_6 + L_{11}^m T_5 + 1)^2}, \tag{E.8.b}$$

where:

$$T_1 = \sin(\phi_{MOB}^W)\sin(\theta_{MOB}^W)\sin(\psi_{MOB}^W)$$

$$T_2 = -\cos(\phi_{MOB}^W)\sin(\psi_{MOB}^W) + \sin(\phi_{MOB}^W)\sin(\theta_{MOB}^W)\cos(\psi_{MOB}^W)$$

$$T_3 = \cos(\phi_{MOB}^W)\sin(\theta_{MOB}^W)\sin(\psi_{MOB}^W)$$

$$T_4 = \sin(\phi_{MOB}^W)\sin(\psi_{MOB}^W) + \cos(\phi_{MOB}^W)\sin(\theta_{MOB}^W)\cos(\psi_{MOB}^W)$$

$$T_5 = -\sin(\theta_{MOB}^W)x_n^{MOB} + \cos(\theta_{MOB}^W)\sin(\psi_{MOB}^W)y_n^{MOB} + \cos(\theta_{MOB}^W)\cos(\psi_{MOB}^W)z_n^{MOB} + z_{MOB}^W$$

$$T_6 = \sin(\phi_{MOB}^W)\cos(\theta_{MOB}^W)x_n^{MOB} + (\cos(\phi_{MOB}^W)\cos(\psi_{MOB}^W) + T_1)y_n^{MOB} + T_2 z_n^{MOB} + y_{MOB}^W$$

$$T_7 = \cos(\phi_{MOB}^W)\cos(\theta_{MOB}^W)x_n^{MOB} + (-\sin(\phi_{MOB}^W)\cos(\psi_{MOB}^W) + T_3)y_n^{MOB} + T_4 z_n^{MOB} + x_{(MOB)W}$$

$$T_8 = T_4 y_n^{MOB} + (\sin(\phi_{MOB}^W)\cos(\psi_{MOB}^W) - T_3)z_n^{MOB}$$

$$T_9 = T_2 y_n^{MOB} + (-\cos(\phi_{MOB}^W)\cos(\psi_{MOB}^W) - T_1)z_{(n)MOB}$$

$$T_{10} = \cos(\theta_{MOB}^W)\cos(\psi_{MOB}^W)y_n^{MOB} - \cos(\theta_{MOB}^W)\sin(\psi_{MOB}^W)z_n^{MOB}$$

$$T_{11} = L_1^m T_7 + L_2^m T_6 + L_3^m T_5 + L_4^m$$

$$T_{12} = L_5^m T_7 + L_6^m T_6 + L_7^m T_5 + L_8^m$$

$$\frac{\partial u_n^m}{\partial \theta_{MRB}^W} = \frac{(L_1^m T_{16} + L_2^m T_{17} + L_3^m T_{18})}{(L_9^m T_{15} + L_{10}^m T_{14} + L_{11}^m T_{13} + 1)} - \frac{T_{19}(L_9^m T_{16} + L_{10}^m T_{17} + L_{11}^m T_{18})}{(L_9^m T_{15} + L_{10}^m T_{14} + L_{11}^m T_{13} + 1)^2}, \quad \text{(E.9.a)}$$

$$\frac{\partial u_n^m}{\partial \theta_{MRB}^W} = \frac{(L_5^m T_{16} + L_6^m T_{17} + L_7^m T_{18})}{(L_9^m T_{15} + L_{10}^m T_{14} + L_{11}^m T_{13} + 1)} - \frac{T_{20}(L_9^m T_{16} + L_{10}^m T_{17} + L_{11}^m T_{18})}{(L_9^m T_{15} + L_{10}^m T_{14} + L_{11}^m T_{13} + 1)^2}, \quad \text{(E.9.b)}$$

where:

$$T_{13} = -\sin(\theta_{MRB}^W) x_n^{MRB} + \cos(\theta_{MRB}^W)\sin(\psi_{MRB}^W) y_n^{MRB} + \cos(\theta_{MRB}^W)\cos(\psi_{MRB}^W) z_n^{MRB} + z_{MRB}^W$$

$$\begin{aligned} T_{14} = {} & \sin(\phi_{MOB}^W)\cos(\theta_{MOB}^W) x_n^{MOB} \\ & + (\cos(\phi_{MOB}^W)\cos(\psi_{MOB}^W) + \sin(\phi_{MOB}^W)\sin(\theta_{MOB}^W)\sin(\psi_{MOB}^W)) y_n^{MOB} \\ & + (-\cos(\phi_{MOB}^W)\sin(\psi_{MOB}^W) + \sin(\phi_{MOB}^W)\sin(\theta_{MOB}^W)\cos(\psi_{MOB}^W)) z_n^{MOB} \\ & + y_{MOB}^W \end{aligned}$$

$$\begin{aligned} T_{15} = {} & \sin(\phi_{MOB}^W)\cos(\theta_{MOB}^W) x_n^{MOB} \\ & + (-\sin(\phi_{MOB}^W)\cos(\psi_{MOB}^W) + \cos(\phi_{MOB}^W)\sin(\theta_{MOB}^W)\sin(\psi_{MOB}^W)) y_n^{MOB} \\ & + (\sin(\phi_{MOB}^W)\sin(\psi_{MOB}^W) + \cos(\phi_{MOB}^W)\sin(\theta_{MOB}^W)\cos(\psi_{MOB}^W)) z_n^{MOB} \\ & + x_{MOB}^W \end{aligned}$$

$$\begin{aligned} T_{16} = {} & \cos(\phi_{MOB}^W)\sin(\theta_{MOB}^W) x_n^{MOB} + \cos(\phi_{MOB}^W)\cos(\theta_{MOB}^W)\sin(\psi_{MOB}^W) y_n^{MOB} \\ & + \cos(\phi_{MOB}^W)\cos(\theta_{MOB}^W)\cos(\psi_{MOB}^W) z_n^{MOB} \end{aligned}$$

$$\begin{aligned} T_{17} = {} & -\sin(\phi_{MOB}^W)\sin(\theta_{MOB}^W) x_n^{MOB} + \sin(\phi_{MOB}^W)\cos(\theta_{MOB}^W)\sin(\psi_{MOB}^W) y_n^{MOB} \\ & + \sin(\phi_{MOB}^W)\cos(\theta_{MOB}^W)\cos(\psi_{MOB}^W) z_n^{MOB} \end{aligned}$$

$$T_{18} = \sin(\theta_{MOB}^W) x_n^{MOB} - \sin(\theta_{MOB}^W)\sin(\psi_{MOB}^W) y_n^{MOB} - \sin(\theta_{MOB}^W)\cos(\psi_{MOB}^W) z_n^{MOB}$$

$$T_{19} = L_1^m T_3 + L_2^m T_2 + L_3^m T_1 + L_4^m$$

$$T_{20} = L_5^m T_3 + L_6^m T_2 + L_7^m T_1 + L_8^m$$

$$\frac{\partial u_n^m}{\partial \phi_{MRB}^W} = \frac{L_1^m(-T_{28}) + L_2^m T_{30}}{L_9^m T_{23} + L_{10}^m T_{22} + L_{11}^m T_{21} + 1} - \frac{T_{12}(L_9^m(-T_{28}) + L_{10}^m T_{30})}{(L_9^m T_{23} + L_{10}^m T_{22} + L_{11}^m T_{21} + 1)^2}, \quad \text{(E.10.a)}$$

$$\frac{\partial v_n^m}{\partial \phi_{MRB}^W} = \frac{L_5^m(-T_{28}) + L_6^m T_{30}}{L_9^m T_{31} + L_{10}^m T_{29} + L_{11}^m T_{27} + 1} - \frac{T_{13}(L_9^m(-T_{28}) + L_{10}^m T_{30})}{(L_9^m T_{31} + L_{10}^m T_{29} + L_{11}^m T_{27} + 1)^2}, \quad \text{(E.10.b)}$$

where:

$$T_{21} = (\sin(\phi_{MRB}^W)\sin(\psi_{MRB}^W) + \cos(\phi_{MrB}^W)\sin(\theta_{MRB}^W)\cos(\psi_{MRB}^W))z_n^{MRB}$$

$$T_{22} = (-\sin(\phi_{MRB}^W)\cos(\psi_{MRB}^W) + \cos(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\sin(\psi_{MRB}^W))y_n^{MRB}$$

$$T_{23} = \cos(\phi_{MRB}^W)\sin(\theta_{MRB}^W)x_n^{MRB}$$

$$T_{24} = \sin(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\cos(\psi_{MRB}^W)$$

$$T_{25} = \sin(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\sin(\psi_{MRB}^W)$$

$$T_{26} = \sin(\phi_{MRB}^W)\sin(\theta_{MRB}^W)x_n^{MRB}$$

$$T_{27} = -\sin(\theta_{MRB}^W)x_n^{MRB} + \cos(\theta_{MRB}^W)\sin(\psi_{MRB}^W)y_n^{MRB} + \cos(\theta_{MRB}^W)\cos(\psi_{MRB}^W)z_n^{MRB} + z_{MRB}^W$$

$$T_{28} = T_{26} + (\cos(\phi_{MRB}^W)\cos(\psi_{MRB}^W) + T_{25})y_n^{MRB} + (-\cos(\phi_{MRB}^W)\sin(\psi_{MRB}^W) + T_{24})z_n^{MRB}$$

$$T_{29} = T_{28} + y_{MRB}^W$$

$$T_{30} = T_{23} + T_{22} + T_{21}$$

$$T_{31} = T_{30} + x_{MRB}^W$$

$$T_{32} = L_1^m T_{31} + L_2^m T_{29} + L_3^m T_{27} + L_4^m$$

$$T_{33} = L_5^m T_{31} + L_6^m T_{29} + L_7^m T_{27} + L_8^m$$

$$\frac{\partial u_n^m}{\partial x_{MRB}^W} = \frac{L_1^m}{L_9^m T_{36} + L_{10}^m T_{35} + L_{11}^m T_{34} + 1} - \frac{(L_1^m T_{36} + L_2^m T_{35} + L_3^m T_{36} + L_4^m) L_9^m}{(L_9^m T_{36} + L_{10}^m T_{35} + L_{11}^m T_{34} + 1)^2} , \quad \text{(E.11.a)}$$

$$\frac{\partial v_n^m}{\partial x_{MRB}^W} = \frac{L_5^m}{L_9^m T_{36} + L_{10}^m T_{35} + L_{11}^m T_{34} + 1} - \frac{(L_5^m T_{36} + L_6^m T_{35} + L_7^m T_{34} + L_8^m) L_9^m}{(L_9^m T_{36} + L_{10}^m T_{35} + L_{11}^m T_{34} + 1)^2} , \quad \text{(E.11.b)}$$

$$\frac{\partial u_n^m}{\partial y_{MRB}^W} = \frac{L_2^m}{L_9^m T_{36} + L_{10}^m T_{35} + L_{11}^m T_{34} + 1} - \frac{(L_1^m T_{36} + L_2^m T_{35} + L_3^m T_{36} + L_4^m) L_{10}^m}{(L_9^m T_{36} + L_{10}^m T_{35} + L_{11}^m T_{34} + 1)^2} , \quad \text{(E.12.a)}$$

$$\frac{\partial v_n^m}{\partial y_{MRB}^W} = \frac{L_6^m}{L_9^m T_{36} + L_{10}^m T_{35} + L_{11}^m T_{34} + 1} - \frac{(L_5^m T_{36} + L_6^m T_{35} + L_7^m T_{34} + L_8^m) L_{10}^m}{(L_9^m T_{36} + L_{10}^m T_{35} + L_{11}^m T_{34} + 1)^2} , \quad \text{(E.12.b)}$$

$$\frac{\partial u_n^m}{\partial z_{MRB}^W} = \frac{L_3^m}{L_9^m T_{36} + L_{10}^m T_{35} + L_{11}^m T_{34} + 1} - \frac{(L_1^m T_{36} + L_2^m T_{35} + L_3^m T_{36} + L_4^m) L_{11}^m}{(L_9^m T_{36} + L_{10}^m T_{35} + L_{11}^m T_{34} + 1)^2} , \quad \text{(E.13.a)}$$

$$\frac{\partial v_n^m}{\partial z_{MRB}^W} = \frac{L_7^m}{L_9^m T_{36} + L_{10}^m T_{35} + L_{11}^m T_{34} + 1} - \frac{(L_5^m T_{36} + L_6^m T_{35} + L_7^m T_{34} + L_8^m) L_{11}^m}{(L_9^m T_{36} + L_{10}^m T_{35} + L_{11}^m T_{34} + 1)^2} , \quad \text{(E.13.b)}$$

where:

$$T_{34} = -\sin(\theta_{MRB}^W) x_n^{MRB} + \cos(\theta_{MRB}^W)\sin(\psi_{MRB}^W) y_n^{MRB} + \cos(\theta_{MRB}^W)\cos(\psi_{MRB}^W) z_n^{MOB} + z_{MRB}^W$$

$$\begin{aligned} T_{35} = {} & \sin(\phi_{MRB}^W)\cos(\theta_{MRB}^W) x_n^{MRB} \\ & + (\cos(\phi_{MRB}^W)\cos(\psi_{MRB}^W) + \sin(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\sin(\psi_{MRB}^W)) y_n^{MRB} \\ & + (-\cos(\phi_{MRB}^W)\sin(\psi_{MRB}^W) + \sin(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\cos(\psi_{MrB}^W)) z_n^{MRB} \\ & + y_{MRB}^W \end{aligned}$$

$$\begin{aligned} T_{36} = {} & \cos(\phi_{MRB}^W)\cos(\theta_{MRB}^W) x_n^{MRB} \\ & + (-\sin(\phi_{MRB}^W)\cos(\psi_{MRB}^W) + \cos(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\sin(\psi_{MRB}^W)) y_n^{MRB} \\ & + (\sin(\phi_{MRB}^W)\sin(\psi_{MRB}^W) + \cos(\phi_{MRB}^W)\sin(\theta_{MRB}^W)\cos(\psi_{MRB}^W)) z_n^{MRB} \\ & + x_{MRB}^W \end{aligned}$$

Appendix F: Detection and Rejection of False Data

This appendix presents an algorithm implemented in the Argo Program that deals with the combined problems of correspondence, data validation and noise rejection. This algorithm is based on a series of validation tests and if an observed and estimated marker pair pass all of the tests then they are accepted as being valid. This method is slow but function reliably. Future versions of the program will have to include improvements to this section of code to improve performance to allow it to be used in real-time.

F.1 Nomenclature

NDOF	number of degrees of freedom (default = 6)
NORD	numerical order of model (default = 3)
M	number of cameras
N	number of markers on target frame
NM	number of observed markers in a image
q	number of state variables = NORD · NDOF
r	number of estimated observations = $2 \cdot M \cdot N$
k	discrete time index ($k \in$ integer set)
C_k	linearized observation model [r x q]
R_k	variance matrix of observation noise vector [r x r]
$\tilde{g}_k$	incoming image coordinates vector [NM x 1]
$\hat{g}_k^{\dagger}$	estimated incoming image coordinates vector [r x 1]

F.2 Validation Test (Detection)

The validation test tries to pair up the estimated marker with the corresponding observed marker. This process should detect and reject phantom observed markers and estimated markers that do not appear in the observed image.

Correspondence lists for both the estimated image coordinates and the observed image coordinates are initially set to zero, 0, indicated no pairing for the corresponding image coordinates. For

an observed marker paired with an estimated marker the observed data list will have the identification number of the estimated marker and the estimated data list will have the position of the observed marker in the list of observations made for that image and the camera identification number that the image was captured with. In the event that either and observed or estimated marker is rejected the entry in the list corresponding to that marker is set to negative one, -1. An estimated marker can be rejected if it is paired with two or more observed markers, because of an overlap, or it is not paired at all with an observed marker, as a result of occlusion. An observed marker can be rejected if it has no corresponding estimate, because it is a phantom marker, or if it and another marker share space in the tolerance circle of an estimated marker. This can occur if two markers appear to be colliding an overlap in the image but not necessarily in test volume space. Rather than keeping the ambiguous data it was decided to reject all of the data associated with an overlap. This is one reason why it is necessary to have more then four markers on the target making the system overdetermined.

Each estimated marker is compared with each observed marker for a given image. This necessary to check for overlap, phantom markers, and different ordering of image coordinates in the observed data vector with that of the estimated data vector. Different ordering can result from the left to right top to bottom stream that the observed image data is captured in. If for example a marker appeared on the left side of the image one line below another marker to the right of it would second in the sequence. If the estimation places it on the same line as the one on the right then it will precede the one to the right in the estimation stream. The difference of one line in a 420 line image is relatively small and falls within the estimation tolerance.

For each estimated marker the radial distance is computer for each observed marker in the image. This radial distance is compared with a setable radial tolerance. If the radial distance is less than the radial tolerance then it is considered a matching pair. Additional tests are used to determine whether this is a valid pair.

IF radial distance < radial tolerance THEN

- IF no existing match has been made for either the current observed or estimated markers THEN accept both as a pair and mark the observed list with the identification number of estimated marker and the estimated list with the identification number of the observed marker.
- ELSE IF the current estimated marker is already rejected THEN reject the current observed marker.
- ELSE IF the current observed marker is already rejected THEN reject the current estimated marker.
- ELSE IF both the current observed and estimate markers are already paired with different partners THEN reject both the current observed and estimated markers along with the their already existing partners.
- ELSE IF the current estimated marker is already paired with another observed marker THEN reject the current estimate and both the current observed and existing observed markers.
- ELSE IF the current observed marker is already paired with another estimated marker THEN reject the current observed marker and both the current and existing estimated markers.

END

F.3 Removing false data (Rejection)

Once the incoming, $\tilde{g}_k$, and estimated, $\hat{g}_k^{\dagger}$, data vectors have been tested any image coordinate pair from either data set that failed the battery of validity tests is removed from the data vectors and the filter is scaled down appropriately. If an estimated marker is not validated by pairing it with an observed marker then:

- the corresponding image coordinate pair is removed from $\hat{g}_k^{\dagger}$.
- the corresponding rows and columns in the observation covariance model, R_k, are removed.
- the corresponding rows in the linearized observation matrix, C_k, are removed.
- its null identity is removed from the estimation identification list.
- number of estimated markers, r, is decremented by one.

If an observed marker is not validated by pairing it with an estimated marker then:

- the corresponding image coordinate pair is removed from $\tilde{g}_k$.
- its null identity is removed from the observation identification list.
- the number of observed markers, *NM*, is decremented by one.

With the rejected data removed the observed and estimated data vectors should now be the

same length and the Kalman filter observation model is scaled to match. If the dimensions do not match then an error has occurred and the program is terminated. If the dimensions of the vectors match then the observed data vectors is sorted into ascending order using two keys view number and marker number, from the observed data list. The sort is a bubble sort modified for a twin key sort.